CHALMERS
UNIVERSITY OF TECHNOLOGY

Cochlear™

Design and Simulation of a Microcontroller-Based Class D Power Amplifier for Piezoelectric Actuators in Bone Conduction Hearing Aids

Master's thesis in Biomedical Engineering

ALBIN REMNESJÖ BERGSTRAND
EMIL JOHANSSON

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Design and Simulation of a Microcontroller-Based Class D Power Amplifier for Piezoelectric Actuators in Bone Conduction Hearing Aids

Albin Remnesjö Bergstrand Emil Johansson

Department of Electrical Engineering
Division of Communications, Antennas, and Optical Networks
Chalmers University of Technology
Gothenburg, Sweden 2025

Design and Simulation of a Microcontroller-Based Class D Power Amplifier for Piezoelectric Actuators in Bone Conduction Hearing Aids
Albin Remnesjö Bergstrand, Emil Johansson

Supervisor: Armin Azhirnian, Cochlear
Examiner: Erik Agrell, Department of Electrical Engineering

Master's Thesis 2025
Department of Electrical Engineering
Division of Communications, Antennas, and Optical Networks
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A visual representation of a skull model with a piezoelectric actuator positioned on its side. The actuator symbolizes the core technology of bone conduction and is illustrated transmitting acoustic vibrations in the skull bone, which is represented as musical notes flowing in the background.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Design and Simulation of a Microcontroller-Based Class D Power Amplifier for Piezoelectric Actuators in Bone Conduction Hearing Aids.
Albin Remnesjö Bergstrand, Emil Johansson
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Conventional linear audio power amplifiers have been and continue to be popular in audio applications. In recent years, digital Class D switching power amplifiers have increased in popularity, as the demand for smaller and more efficient electronic devices has surged. Continuous technological advances have further improved audio quality and allowed for smaller and more efficient Class D amplifiers. One audio technology that uses Class D amplifiers is bone conduction hearing aids. These systems generally consist of an extrinsic microphone and sound processor, connected to a percutaneous device with the purpose of generating sound through vibrations to the skull bone. Historically, bone conduction devices have been designed and operated based on electromagnetic principles using electromagnetic actuators. Over time, a more efficient type of actuator is increasing in popularity, one that utilizes the piezoelectric force. Low-voltage Class D amplifiers are often only available in integrated circuits (ICs), which take a long time to develop and can be a bottleneck in the process. This thesis explores an alternative approach by developing a design and simulation framework for a low-voltage Class D power amplifier using discrete components to drive a piezoelectric actuator load. The following approach aims to reduce the development time while showcasing the feasibility of constructing low-voltage Class D power amplifiers utilizing discrete components for piezoelectric actuator loads, achieving performance comparable to integrated solutions. A Class D amplifier, using a H-bridge power stage, with $\Delta\Sigma$ to BD PWM with dead-time control, has been developed in simulation and with PCB design. The amplifier, operating across the audio frequency range (20 Hz to 20 kHz), achieved a peak SNR of 69.32 dB and a THD+N peak of 0.15% when driven with a 0.658 V_{RMS} sinusoidal input with a fundamental frequency of 4410 Hz. A peak apparent efficiency of 86.4% was observed with an input voltage of 2.44 V_{RMS}. Compared to IC designs reported in technical literature and found in commercial products, the results are promising. Thus, the conclusion is that the idea of designing a low-voltage discrete Class D amplifier for piezoelectric hearing aids is deemed viable. Still, its performance could be greatly improved by incorporating gate drivers, a higher-order $\Delta\Sigma$ modulator, and more refined filter designs.

Keywords: Bone conduction, hearing aid, piezoelement, sigma Delta, Class D amplifier

Acknowledgements

Firstly, we would like to express our sincere gratitude to Cochlear Bone Anchored Solutions AB for providing us with the opportunity to conduct our master's thesis at the company and for their confidence in our abilities. We would like to deeply thank our supervisor at Cochlear, Armin Azhirnian for his invaluable support and knowledge. The project took longer, required more work and effort than expected, and without Armin's knowledge and encouragement, completing this work would not have been possible. We sincerely thank our supervisor at Chalmers University of Technology, Björn Langborn for his assistance in organizing the report and administrative assistance, which have been greatly appreciated. We also would like to thank our examiner and Professor Erik Agrell for valuable insight and support when writing and structuring the thesis. Lastly, we send our greatest gratitude to those who have supported us throughout this journey, colleagues, friends, and family, thanking you for your encouragement and belief in us.

Albin Remnesjö Bergstrand, Emil Johansson, Gothenburg, 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ACHA	Air Conduction Hearing Aid
ADC	Analog-to-Digital Converter
AD PWM	2-level PWM; A indicates the PWM level, D indicates Class D
BD PWM	3-level PWM; B indicates the PWM level, D indicates Class D
BCH	Bone Conduction Hearing
BCHA	Bone Conduction Hearing Aid
BEST	Balanced Electromagnetic Separation Transducer
BTL	Bridge-Tied Load
CMOS	Complementary Metal Oxide Semiconductor
DAC	Digital-to-Analog Converter
DSP	Digital Signal Processing
dB	Decibel
dBFs	Decibel relative to Full Scale
DevKit	Development Kit
EM	Electromagnetic
EV-Kit	Evaluation Board
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
IC	Integrated Circuit
I^2S	Integrated Inter-IC Sound Bus
LSB	Least Significant Bit
MCU	Microcontroller Unit
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MSB	Most Significant Bit
NTF	Noise Transfer Function
OSR	Oversampling Ratio
PC	Polycarbonate
PCB	Printed Circuit Board
PCM	Pulse Code Modulation
PDM	Pulse Density Modulation
PLL	Phase-Locked Loop
PSD	Power Spectral Density

PWM	Pulse Width Modulation
RMS	Root-Mean-Square
SCK	Continuous Serial Clock
SD	Serial Data
SINAD	Signal-to-Noise and Distortion Ratio
SNR	Signal-to-Noise Ratio
SPL	Sound Pressure Level
STF	Signal Transfer Function
THD	Total Harmonic Distortion
THD+N	Total Harmonic Distortion + Noise
WS	Word Select
$\Delta\Sigma$	Delta-Sigma

Nomenclature

Below is the nomenclature of parameters and variables that have been used throughout this thesis.

Parameters

DF	Dissipation factor
M	Sum iteration number for THD and THD+N
q	Quantization step size
T	One period
ω	Angular frequency, $2\pi f$

Variables

C_{P}	Parasitic capacitor
C_{PZ}	Capacitance of actuator (PZ = piezo)
E_{C}	Energy stored in a parasitic capacitor
e_{q}	Quantization error
f_0	Fundamental frequency of the sinusoidal input
F_{audio}	Input audio frequency
f_{PWM}	PWM clocking frequency
f_{r}	Resonance frequency
f_{s}	Original sampling rate
F_{s}	Final sampling rate after oversampling
F_{SW}	Switching frequency of the amplifier
I_{g}	Gate current
$\hat{I}_{\mathrm{o}}$	Output peak current
$I_{\mathrm{o,RMS}}$	Output RMS current

I_{rrm}	Maximum body-diode reverse-recovery current
I_{Q}	Quiescent current
$i(t)$	Instantaneous output current
N	Number of bits
P_0	Power of the fundamental frequency component
P_{BD}	Body-diode power losses
P_{C}	Average consumed power in a parasitic capacitance
P_{CL}	Conduction power losses
P_{dt}	Dead time power losses
P_{FILT}	Power losses in the output filter
P_{in}	Input power
P_{loss}	Power losses in the amplifier for a resistive load
$P_{\mathrm{loss,adj}}$	Total power loss adjusted for the model in this work
P_{n}	Power of noise in the bandwidth of interest
P_{o}	Output power
$P_{\mathrm{o,app}}$	Apparent output power
P_{reac}	Reactive power
P_x	Power of harmonics in the bandwidth of interest
P_{rr}	Reverse-recovery charge power loss
P_{SW}	Switching power losses
P_{trans}	Transition power loss
P_{Q}	Quiescent power losses
Q	Quality factor
$Q(x)$	Quantized output
Q_{g}	Total gate charge
q	Quantization step size
r	Exponent of the oversampling rate
$R_{\mathrm{DS(on)}}$	On-state drain–source resistance
t_{dt}	Dead-time implemented to avoid Shoot-through current
t_{rr}	Body-diode reverse-recovery time
t_{switch}	Switching time (time to charge the MOSFET gate)
t_{trans}	Transition time of the MOSFET
$v(t)$	Instantaneous output voltage
V_{GS}	MOSFET Gate-Source voltage

V_{i}^{2}	RMS input voltage squared
V_{CP}	Voltage across a parasitic capacitor
$\hat{V}_{\mathrm{i}}$	Input peak voltage
$V_{\mathrm{o,RMS}}$	RMS output voltage
V_{n}	RMS noise voltage within the bandwidth of interest
V_{DD}	Supply voltage
V_{GS}	Gate–source voltage
V_{rail}	Rail voltage connected to PMOS1 and PMOS2
V_{SD}	Body-diode source-to-drain voltage
X_{C}	Capacitive reactance
X_{L}	Inductive reactance
x	Input signal to quantizer
σ_e^2	Average quantization noise power
σ_x^2	Average power of quantized signal
$\lvert Z_F\rvert \cos(\phi)$	Resistive part of the filter at audio frequency f_{PWM}
ϕ	Phase shift between current and voltage

Contents

List of Figures

List of Tables

1

Introduction

1.1 Background

The human ear is a complex organ that serves multiple functions and plays a key role in daily life. It helps facilitate communication and shapes the way we interact with the world. It consists of three different regions: the external ear, the middle ear, and the inner ear [2]. These three regions have different functions but work together to process sound and convert them into neural impulses. These impulses travel to the brain, where they are processed and interpreted as sound. When discussing the sense of hearing, it is divided into two distinct components: bone conduction (BC) and air conduction (AC). In AC hearing (ACH), sound waves enter the outer ear, travel through the middle ear, and cause the eardrum to vibrate [3]. The vibrations are further transmitted into the inner ear until they reach the cochlea, which in turn converts the vibrations into nerve signals. The second component, BC hearing (BCH), refers to the auditory response of vibrations through the human skull. These vibrations continue to travel through the skull until they reach the inner ear, where they stimulate the cochlea and result in the sensation of hearing.

When an individual has a hearing threshold of 25 dB or higher, he or she is diagnosed with hearing loss, which can severely affect the quality of life of that individual [4, 5]. In general, there are three main types of hearing loss: sensorineural, conductive, and mixed hearing loss [6]. Sensorineural loss affects the inner ear or nerve signals, while conductive hearing loss affects the outer and middle ear. Mixed hearing loss is a mixture of the two. AC does not work as intended for a person with conduction hearing loss, which results in a higher hearing threshold for sound traveling through the air. Patients diagnosed with conductive or mixed hearing loss can achieve an improved hearing threshold using a bone conduction hearing aid (BCHA) that can be implanted or extrinsic [7, 8]. Implanted hearing aids can be percutaneous or transcutaneous. Percutaneous devices are surgically implanted devices directly attached to the temporal bone, with an abutment protruding through the skin [9]. In transcutaneous devices, the implanted part and the external processor of the hearing aid are connected by magnetic discs, hence no skin penetrating abutment is needed [10, 11].

1.2 Research Gap and Motivation

Today, modern BCHAs use integrated circuits (IC), such as complementary metal oxide semiconductor (CMOS) technology, to power their speakers due to their low power consumption. The external component of the hearing aid, located outside the body, is powered by a battery. Reducing power consumption of the electronics is important to ensure a long battery life and thus improve the quality of life of users. Designing efficient power amplifiers in BCHAs with low power consumption is a complex task that demands a significant time, effort, and high development costs. Once developed, ICs offer several advantages, such as the capability for mass production at a low cost per unit, compact size, and energy efficiency. Although power amplifiers designed with discrete components might not reach the same level of performance as ICs, they offer a shorter integration timeline and are more cost-effective during the early design phases. If the power amplifier stage can be built using discrete power-MOSFETs while still delivering performance close to that of an integrated circuit, it will serve as an enabler for many products.

To the best of the authors' knowledge, no Class D amplifiers for piezoelectric actuators designed with discrete components are found in either commercial products or technical literature. Even Class D amplifiers for piezoelectric actuators with integrated solutions are scarce in technical literature, and only in the last couple of years some literature has been published on the topic. This gap highlights the need for further research into efficient, high-performance power amplifiers using discrete component solutions for bone conduction hearing aids.

1.3 Aim and Research Question

The aim of this thesis is to develop a design and simulation framework of a digital Class D power amplifier for piezoelectric actuators in BCHAs, using off-the-shelf discrete components. The objective is to achieve performance levels close to those of ICs in modern BCHAs by achieving low total harmonic distortion (THD) and total harmonic distortion plus noise (THD+N), high apparent efficiency, and minimized power consumption. This approach seeks to provide a cost-effective and faster alternative to the traditional IC development process, filling a gap in the current market. Ultimately, improving the quality of life of users by prolonging battery life and improving audio performance.

To achieve this aim, the thesis will address the following question:

I. Is it feasible to design and implement a microcontroller-based digital Class D power amplifier using discrete components for piezoelectric actuators in BCHAs, and to achieve performance comparable to integrated solutions?

1.4 Objectives

The thesis aims to achieve the following objectives:

I. Design and simulate a second order Delta Sigma ($\Delta\Sigma$) modulator that can be implemented for real-time audio processing on a microcontroller with audio input using the I^2S (Inter-IC Sound) protocol
II. Implement an efficient $\Delta\Sigma$-to PWM conversion method for driving discrete power MOSFETs in the output stage of a Class D power amplifier
III. Design a custom-made PCB for the proposed digital power amplifier system using off-the-shelf components
IV. Optimize the second order $\Delta\Sigma$ modulator by experimenting with the bit depth in the quantized output, to maximize signal-to-noise ratio (SNR) and minimize noise
V. Experiment with dead-time control to find an optimal dead-time that minimizes current spikes and prevents short circuits
VI. Minimize THD and THD+N in the digital Class D audio amplifier using test signals within the 20 Hz to 20 kHz audio frequency range
VII. Achieve a quiescent power consumption of < 1 mW
VIII. Ensure apparent power amplifier efficiency of $\geq 80\%$
IX. Provide recommendations for further development and potential improvements

1.5 Delimitations

This section outlines the boundaries established for this thesis.

- The audio input will be generated in simulation environment in MATLAB as an exclusively digital signal, with a sampling frequency of 44.1 kHz, and with 1000 samples represented in the form of a 16-bit quantized sinusoidal signal. The different frequencies chosen for the simulations are integer multiples of 44.1 Hz, ensuring that the waveform completes an integer number of cycles in order to avoid spectral leakage in the simulation environment.
- Only off-the-shelf microcontrollers for driving the Class D amplifier will be investigated.
- The modulation techniques will be limited to $\Delta\Sigma$ modulation and conversion to PWM for signal processing due to compatibility with digital systems and effective noise shaping characteristics.
- The PCB will initially be designed using KiCad software and later printed, but not soldered as it is beyond the scope of this work.
- Integrating the digital audio input from the sound card via I2s into the microcontroller and testing the H-bridge circuitry required expertise and time that exceeded the scope of this thesis. Therefore, they were designated for future work.
- The piezoelectric actuator used as the output load will not be tested practically due to time constraints. Instead, its behavior will only be analyzed through simulations, where it will be modeled as a capacitor together with a series resistance.

2

Fundamentals of Audio Amplification in BCHAs

This chapter begins with a brief overview of the key components in a typical digital BCHA system, illustrated in Fig. 2.1 and highlights the specific components examined in this thesis. Next, the most widely used bone vibrators for BCHAs will be covered, including their key characteristics and operational principles. Then, the fundamental performance indicators of audio amplification will be introduced.

2.1 Overview of Digital BCHA Systems

A generic block diagram of a digital BCHA is illustrated in Fig. 2.1. The hearing aid consists of a microphone, preamplifier, analog-to-digital converter (ADC), digital signal processor, digital-to-analog converter (DAC), a power amplifier, and a receiver. For air conduction hearing aids (ACHAs), the receiver is in the form of a speaker, whereas for those operating on the principle of BC, the receiver is in the form of a transducer or actuator. In this work, the focus is only on the components within the orange sub-block in Fig. 2.1.

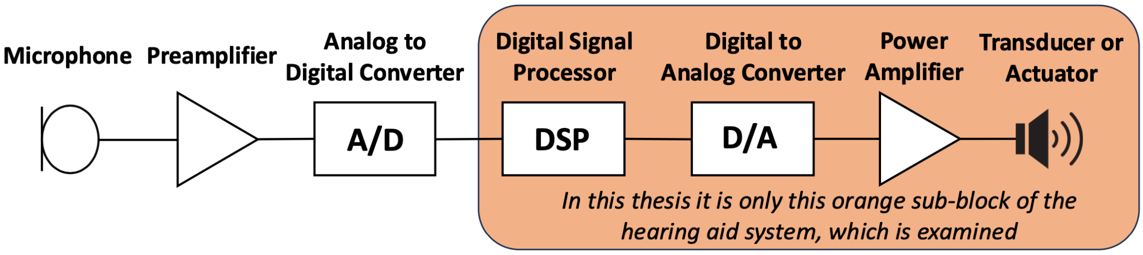


Figure 2.1: A generic digital BCHA system.

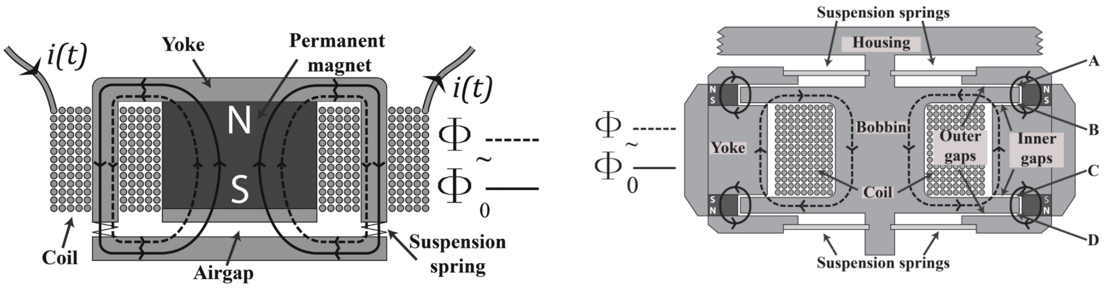
2.2 Vibration Technologies in BCHAs

This section provides a brief overview of the most commonly used vibrators for BCHAs, focusing on electromagnetic transducers and piezoelectric ceramic actuators.

2.2.1 Electromagnetic Transducers

Over the years the preferred transducer for hearing aids has been the electromagnetic (EM) transducer, due to its efficient mechanism and ability to generate high levels of sound pressure [12, 13]. Fig. 2.2 illustrates two widely used types of EM transducers: the B71 and the B81. The transducer consists of a voice coil, a magnet, and an acoustic cavity. When an electrical current is applied to the coils, an electromagnetic field is generated, in turn generating static magnetic flux in the air gaps. This causes a displacement in the acoustic diaphragm, creating vibrations or sound [14].

However, there are known disadvantages with EM speakers, such as the need of a large form factor to generate a high sound pressure level (SPL). This is primarily due to their low impedance value, typically ranging from 4 to 32 Ω [12]. As a result, the amplifier must supply large electrical current through the voice coil to generate a high SPL. This high current demand leads to increased power consumption, which is a problem for battery powered devices. Thus, regardless how efficiently you design the amplifying circuit, the EM transducer inherently limits the battery life.



(a) **(b)**

Figure 2.2: **(a)** Cross-sectional view of the B71 BC transducer. **(b)** Cross-sectional view of the B81 or Balanced Electromagnetic Transducer (BEST). Both images from [14]. Allowed by author to use.

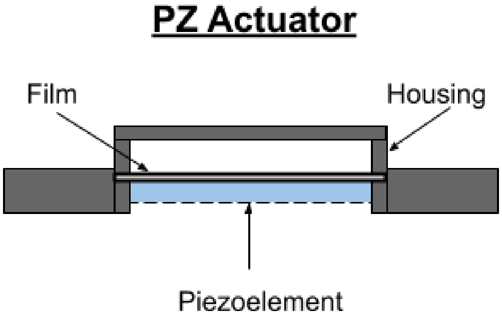


Figure 2.3: Structure of a generic piezo actuator

2.2.2 Piezoelectric Ceramic Actuators

In recent years, the piezoelectric ceramic actuator has gained significant attention, due to its power efficiency, low weight, and its thin construction, allowing for more compact speaker designs [15].

The actuator consists of a piezoelectric ceramic element (piezoelement), which is mounted on an acoustic diaphragm or a film, and a housing that functions as an overall enclosure containing the speaker components. The generic structure of a piezoelectric speaker is illustrated in Fig. 2.3. Common construction materials are polycarbonate (PC) for the housing, plastic resin or metal for the film, and lead zirconate titanate (PZT) for the piezoelement [12]. When the piezoelement receives a voltage, the ceramic membrane deforms and begins to bend upward and downward depending on the voltage applied, which in turn causes the film to vibrate and generate sound waves [15, 16].

2.3 Performance Measures of Audio Amplifiers

2.3.1 Dynamic Range

The dynamic range is a very important indicator in audio systems, as it represents the range between the loudest signal and the quietest representable signal, highlighting the ability of the audio amplifier system to capture both very quiet and loud sounds. For a 16-bit digital system, the smallest representable value in the system is 1 least significant bit (1 LSB), and the maximum value 0 dBFS corresponds to all bits set to '1'. The theoretical dynamic range for an N-bit audio signal is given by the formula:

$$\text{Dynamic Range (dB)} = 6.02N, \quad (N = \text{number of bits})$$

Thus, the theoretical dynamic range for a 16-bit system is 96.32 dB, which means that the maximum value is 96.32 dB above the quietest value.

2.3.2 Signal-to-Noise Ratio

The SNR is an indicator used to express the ratio between the signal and the noise produced by the amplifier. Sources of noise can include quantization error, thermal noise from the circuit components, and radio frequency interference, among others. If the impedance is constant for both the signal and the noise the SNR can be expressed as:

$$\mathrm{SNR} = 10\mathrm{log}\frac{P_0}{P_\mathrm{n}} = 20\ \mathrm{log}\frac{V_0}{V_\mathrm{n}} \tag{2.1}$$

where P_0 and V_0 represent the power and output voltage of the fundamental frequency component of the audio signal, respectively. P_n and V_n denote the power and voltage output noise within the audio spectrum 20 to 20 000 Hz. When measuring the SNR, only the ratio of power of the fundamental and the power of the noise is considered, with the power of the harmonics being excluded.

2.3.3 Distortion Metrics

The audio quality is of high importance in hearing aid systems. To reproduce real-world sound accurately, it is very important to minimize the distortion. Distortion in audio systems implies an alteration of the output signal from the original input [17], and it is typically generated by unintended artifacts introduced by components within the audio system. In hearing aids, there are two standard audio quality metrics, THD and THD+N. The formula for THD is given by [12]:

$$\mathrm{THD} = \sqrt{\sum_{i=1}^{K} \frac{V_i^2}{V_0^2}} = \sqrt{\sum_{i=1}^{K} \frac{P_i}{P_0}} \tag{2.2}$$

where V_0 is the RMS voltage of the fundamental frequency component, V_i is the RMS voltage of the harmonics. This expression can be reformulated in terms of power where, P_0 is the power of the fundamental frequency, P_i is the power of the harmonics and K is the index of the harmonics up to 20 kHz [12]. In the field of audio quality, THD+N is the most commonly used and reliable indicator for audio quality due to the addition of both harmonic distortion and noise.

The formula for THD+N is given by [12]:

$$\mathrm{THD} + \mathrm{N} = \sqrt{\sum_{i=1}^{K} \frac{V_i^2}{V_0^2} + \frac{V_\mathrm{n}^2}{V_0^2}} = \sqrt{\sum_{i=1}^{K} \frac{P_i}{P_0} + \frac{P_\mathrm{n}}{P_0}} \tag{2.3}$$

where once again V_0 and P_0 are the RMS voltage and the power of the fundamental frequency respectively, while V_n is the RMS voltage of the noise within the bandwidth of interest, usually in the audio spectrum 20 to 20000 Hz. This expression can be written in terms of dB as:

$$\mathrm{THD} + \mathrm{N}_\mathrm{dB} = 20\log_{10}\sqrt{\sum_{i=1}^{K} \frac{V_i^2}{V_0^2} + \frac{V_\mathrm{n}^2}{V_0^2}} = 10\log_{10}\left(\sum_{i=1}^{K} \frac{P_i}{P_0} + \frac{P_\mathrm{n}}{P_0}\right) \tag{2.4}$$

In technical articles and literature another performance metric called signal to noise and distortion ratio (SINAD) is often presented, and is defined as [18]:

$$\mathrm{SINAD_{dB}} = 10\log\sum_{i=1}^{K}\frac{P_0}{P_\mathrm{n}+P_\mathrm{i}} = 20\log\sum_{i=1}^{K}\frac{V_0}{\sqrt{V_\mathrm{n}^2+V_\mathrm{i}^2}} \tag{2.5}$$

where P_0 and V_0 represent the power and voltage of the fundamental signal, respectively. P_n and V_n denote the power and voltage of the noise, while P_i and V_i represent the power and voltage of the signal harmonics. By rearranging the terms in $\mathrm{THD+N_{dB}}$ it is evident that (2.4) and (2.5) are related by a negative sign:

$$\mathrm{SINAD_{dB}} = -\mathrm{THD+N_{dB}}$$

The total harmonic distortion (THD) in commercial BCHAs is typically measured and reported according to established standards such as IEC60118. For BCHAs common test frequencies are 500, 800, 1600 and 3200 Hz for certain sound pressure levels (SPL) [19, 20]. Acceptable (THD) values vary depending on the type of hearing aid used, but a general guideline is that the distortion should not be greater than 10% for a given frequency [21]. For percutaneous BCHAs devices, such as the Cochlear Baha 6 the THD is less than 0.3%, and for transcutaneous BCHAs devices, such as the Oticon Sentio 1 Mini the THD is less than 1% for frequencies (800 to 3200 Hz). In comparison, for BCHAs with piezoelectric loads such as the Cochlear Osia 2, the THD is reported to be less than 5% above 600 Hz.

2.3.4 Efficiency of Switching Audio Amplifiers with Resistive Loads

One of the key performance indicators in switching audio amplifiers with resistive loads is power efficiency, as it represents how much of the supplied energy is converted into audio output and how much of that energy is lost, typically as heat. When measuring power efficiency in a conventional amplifier with a resistive load, efficiency is determined by calculating the ratio of output power to input power (the supply) [16]. The input power P_in is expressed as the sum of the output power P_o and the power lost P_loss:

$$P_\mathrm{in} = P_\mathrm{o} + P_\mathrm{loss} \tag{2.6}$$

This is generally defined by using the average consumed power P_avg over one sinusoidal signal period, given by the voltage $v(t)$ and the current $i(t)$ over one period T.

$$P_\mathrm{avg} = \frac{1}{T}\int_0^T v(t)\,i(t)\,dt = V_\mathrm{o} I_\mathrm{o}\cos(\phi) \tag{2.7}$$

The phase angle ϕ, appearing in the power factor term $\cos(\phi)$ from (2.7), describes the phase relationship between the current (I_o) and the voltage (V_o). Electromagnetic transducers are almost purely resistive, which implies that the current and the voltage are in phase, thus $\phi \simeq 0°$, and $\cos(0°) = 1$.

From (2.6), the power efficiency η for resistive loads can be defined as:

$$\eta = \frac{P_\mathrm{o}}{P_\mathrm{in}} = \frac{P_\mathrm{o}}{P_\mathrm{o} + P_\mathrm{loss}} \tag{2.8}$$

2.3.4.1 Efficiency of Switching Audio Amplifiers with Piezoelectric loads

Since piezoelectric actuators are capacitive in nature, they also consume reactive power, and their current leads their voltage by a phase shift ϕ of almost 90°. This can be seen in Fig. 2.4, which visualizes the output voltage and current across the piezoelectric load in a switching audio amplifier. As a result, the term $\cos(\phi)$ in (2.7) is approximately equal to $\cos(90°) \simeq 0$, making the average power delivered to the load appear significantly lower compared to that of a typical resistive load. Therefore, when (2.7) is used for determining the efficiency for piezoelectric loads, it will seem very low because the power in the circuit is flowing back and forth between the supply and the load. This makes it very difficult to put a meaningful measure on the amplifier's efficiency, and therefore, (2.8) is rarely used as a performance metric in audio amplifiers driving piezoelectric loads.

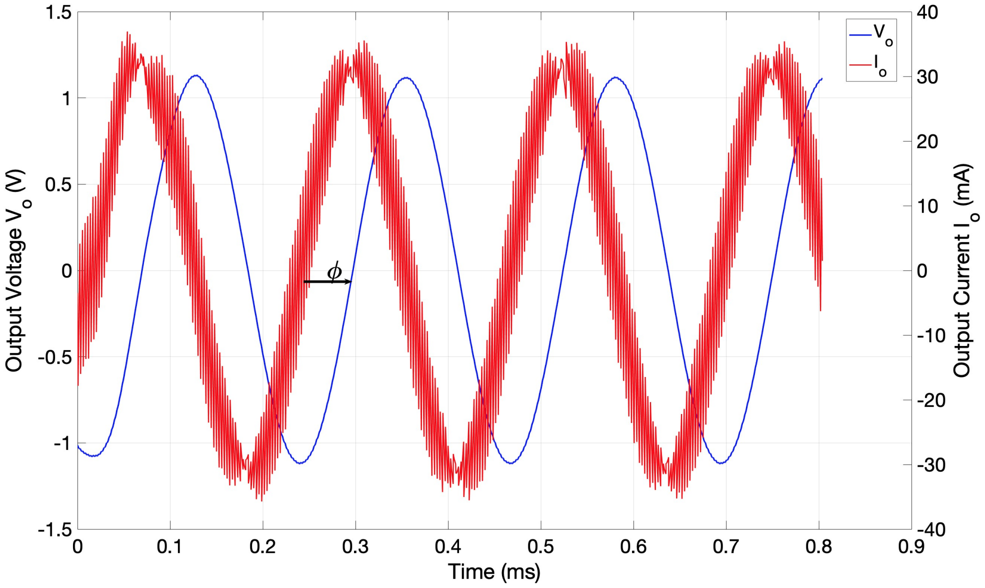


Figure 2.4: Output voltage and current over the piezoelectric actuator in a switching audio amplifier, illustrating the 90° phase shift.

An alternative method for calculating efficiency with loads of capacitive nature is proposed in [12], based on apparent power, P_{app}. Apparent power, represented by the yellow arrow in the power triangle shown in Fig. 2.5, is measured in Volt-Amperes (VA) and represents the vector sum of active power, measured in watts (W), and reactive power, measured in Volt-Amperes reactive (VAr). Based on the relationships in the power triangle, the apparent power can be given by the following:

$$P_{app} = \sqrt{P^2 + P_{\mathrm{reac}}^2} \tag{2.9}$$

where, P is the active power and P_{reac} is the reactive power in the circuit. Apparent power is the total amount of power flowing in an AC circuit and represents the average power without taking the phase into account, expressed as:

$$P_{\mathrm{o,app}} = V_{\mathrm{o,RMS}}\, I_{\mathrm{o,RMS}} \tag{2.10}$$

here $P_{\mathrm{o,app}}$ is the output apparent power and $V_{\mathrm{o,RMS}}$ and $I_{\mathrm{o,RMS}}$ are the RMS output voltage and current over the piezoelectric load. For a load with impedance $|Z_{\mathrm{L}}(\omega)|$, at the fundamental angular frequency ω, $I_{\mathrm{o,RMS}}$ is given by:

$$I_{\mathrm{o,RMS}} = \frac{V_{\mathrm{o,RMS}}}{|Z_{\mathrm{L}}(\omega)|} \tag{2.11}$$

by inserting (2.11) into (2.10) yields:

$$P_{\mathrm{o,app}} = \frac{V_{\mathrm{o,RMS}}^2}{|Z_{\mathrm{L}}(\omega)|} \tag{2.12}$$

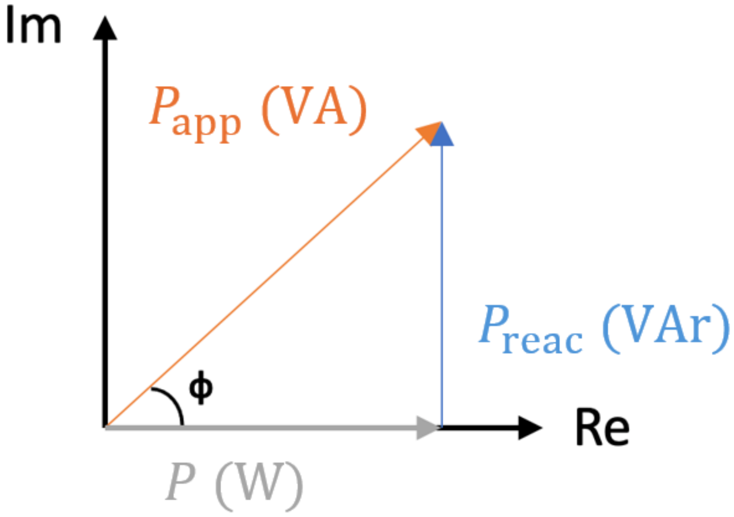


Figure 2.5: Power triangle, showing the relationship between active power, reactive power, and apparent power.

2.3.4.2 Power losses in Switching Audio Amplifiers with Piezoelectric Loads

Based on (2.8), the piezoelectric amplifier's apparent power efficiency can be formulated as [12]:

$$\eta_{\mathrm{PZ}} = \frac{P_{\mathrm{o,app}}}{P_{\mathrm{in}}} = \frac{P_{\mathrm{o,app}}}{P_{\mathrm{o,app}} + P_{\mathrm{loss}}} \tag{2.13}$$

where P_{in} is the average input power consumed by the supply and P_{loss} is the total power loss in the amplifier for a piezoelectric load. Since the average active power P in (2.9) is around 0 for a piezoelectric load the apparent power will consist mainly of the reactive component P_{reac}.

The total power loss P_{loss} in (2.13) can be calculated as [12]:

$$P_{\mathrm{loss}} = P_{\mathrm{Q}} + P_{\mathrm{CL}} + P_{\mathrm{SW}} + P_{\mathrm{BD}} + P_{\mathrm{FILT}} \tag{2.14}$$

here the total power loss is the combination of quiescent power P_{Q}, conductive power P_{CL}, switching power P_{SW}, body-diode power P_{BD}, and piezo output filter power losses P_{FILT}.

The amplifier's quiescent power losses P_{Q} represents the power consumption when the amplifier is in an idle state, receiving only background noise, and is given by:

$$P_{\mathrm{Q}} = V_{\mathrm{DD}}\, I_{\mathrm{Q}} \tag{2.15}$$

where V_{DD} is the supply voltage and I_{Q} is the quiescent current drawn from the supply.

Conduction power loss, P_{CL}, refers to the power dissipated during the brief time period when the MOSFET is conducting and acts like a resistor $R_{\mathrm{DS(on)}}$. This can be calculated as:

$$P_{\mathrm{CL}} = I^2_{\mathrm{o,RMS}}\, R_{\mathrm{DS(on)}} \tag{2.16}$$

in which $I^2_{\mathrm{o,RMS}}$ is the RMS output current and $R_{\mathrm{DS(on)}}$ is the on-state drain–source resistance during conduction.

The switching power losses, P_{SW} is a combination of two components, the charging and discharging of the parasitic capacitances in the MOSFET, and the transition loss during switching. The first component can be estimated by calculating the energy stored in each parasitic capacitor C_{P} of the MOSFET:

$$E_{\mathrm{C}} = \frac{1}{2} C_{\mathrm{P}}\, V^2_{\mathrm{CP}} \tag{2.17}$$

where V_{CP} is the voltage over respective parasitic capacitance. Each parasitic capacitance is charged and discharged every cycle with the switching frequency F_{SW}. Thus, the average consumed power P_{C} in each parasitic capacitance is given by:

$$P_{\mathrm{C}} = F_{\mathrm{SW}}\, E_{\mathrm{C}} = F_{\mathrm{SW}}\, C_{\mathrm{P}}\, V^2_{\mathrm{CP}} \tag{2.18}$$

by summing over every parasitic capacitance in the MOSFET the following expression is obtained:

$$P_{\mathrm{C}} = \sum_i F_{\mathrm{SW}}\, C_{\mathrm{P},i}\, V_{\mathrm{CP}}^2 \tag{2.19}$$

the second component, the transition power loss can be estimated by:

$$P_{\mathrm{trans}} = V_{\mathrm{DD}}\, \hat{I}_{\mathrm{o}}\, 2\, t_{\mathrm{trans}}\, F_{\mathrm{SW}} \tag{2.20}$$

where V_{DD} is the supply voltage connected to the MOSFET, $\hat{I}_{\mathrm{o}}$ is the peak output current through the MOSFET and t_{trans} is the transition time. Since one period has two transitions ON-OFF and OFF-ON a factor of two is needed. By combining the two components (2.19) and (2.20) the complete switching loss formula is yielded [12]:

$$P_{\mathrm{SW}} = \sum_i (F_{\mathrm{SW}}\, V_{\mathrm{CP}}^2\, C_{\mathrm{P},i}) + V_{\mathrm{DD}}\, \hat{I}_{\mathrm{o}}\, 2\, t_{\mathrm{trans}}\, F_{\mathrm{SW}} \tag{2.21}$$

the gate driver power loss P_{g} is given by [22]:

$$P_{\mathrm{g}} = Q_{\mathrm{g}}\, F_{\mathrm{SW}}\, V_{\mathrm{GS}} \tag{2.22}$$

here, Q_{g} is the total gate charge and V_{GS} is the gate to source voltage.

Body-diode losses P_{BD} occur at every switching instance due to the conduction, and reverse recovery charge of the MOSFET's body-diode, located between source and drain. Body diode losses can be estimated by calculating two components: dead time power loss P_{dt}, and reverse recovery loss P_{rr} of the body diode. The first component can be estimated by calculating the losses associated with the reverse-recovery charge of the body diode, given by:

$$P_{\mathrm{rr}} = I_{\mathrm{rrm}}\, t_{\mathrm{rr}}\, V_{\mathrm{SD}}\, F_{\mathrm{SW}} \tag{2.23}$$

where I_{rrm} is the maximum body diode reverse-recovery current, t_{rr} is the body diode reverse recovery time, V_{SD} is the body-diode source-to-drain voltage. The second component, P_{dt} is given by:

$$P_{\mathrm{dt}} = \hat{I}_{\mathrm{o}}\, t_{\mathrm{dt}}\, V_{\mathrm{SD}}\, F_{\mathrm{SW}} \tag{2.24}$$

once again $\hat{I}_o$ is the peak output current and t_{dt} is the implemented dead-time. By combining equations (2.23) and (2.24) the complete body-diode loss formula yields [12]:

$$P_{\mathrm{BD}} = V_{\mathrm{SD}}\, F_{\mathrm{SW}}\, (\hat{I}_{\mathrm{o}}\, t_{\mathrm{dt}} + I_{\mathrm{rrm}}\, t_{\mathrm{rr}}) \tag{2.25}$$

The piezoelectric output filter power loss P_{FILT}, arises from the current ripple in the output signal shown as the blue curve in Fig. 2.6 and the dielectric losses in the piezoelectric actuator at the audio frequency, where the energy is converted into heat in the material. If there had been zero losses in the output signal, the output

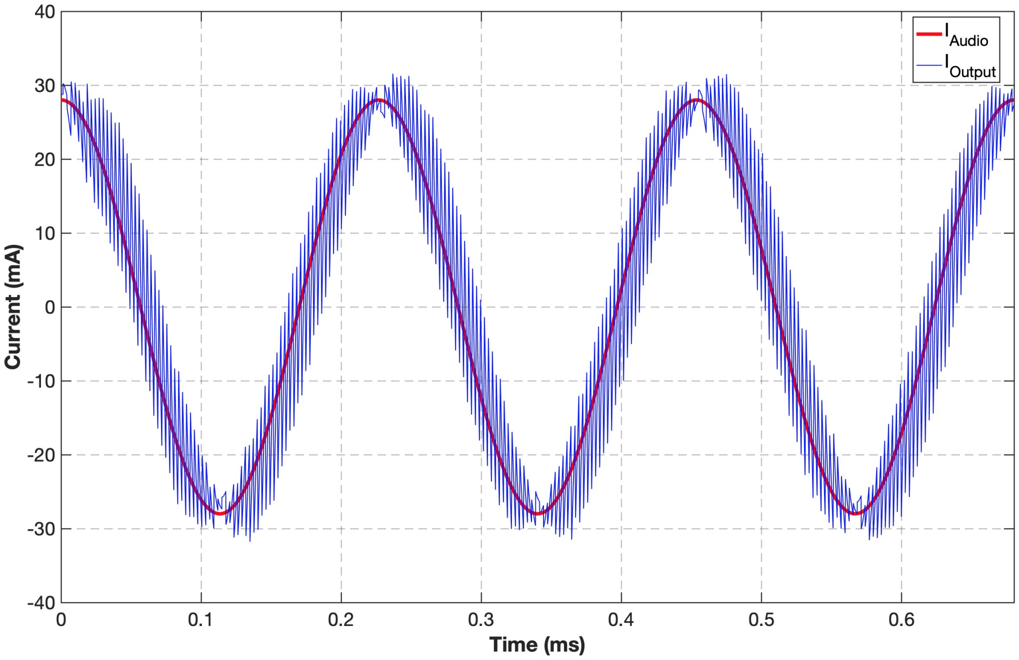


Figure 2.6: Output current trough the piezoelectric actuator together with the ideal audio current.

current would look like the red audio signal in the plot. This loss can be expressed as:

$$P_{\mathrm{FILT}} = I_{\mathrm{o,RMS}}^{2}\,|Z_{\mathrm{L}}(\omega)|\,\cos(\phi) + C_{\mathrm{PZ}}\,V_{\mathrm{o,RMS}}^{2}\,2\pi\,F_{\mathrm{audio}}\,DF \tag{2.26}$$

where $|Z_{\mathrm{F}}(\omega)|\cos(\phi)$ corresponds to the resistive losses of the real part of the output filter at the audio frequency F_{audio}, C_{PZ} represents the capacitance of the piezoelectric actuator, and DF corresponds to the dissipation factor of the specific piezoelectric actuator. Since the output current consists of both the audio frequency component as well as the switching frequency component, it can be formulated as: $I_{\mathrm{o,RMS}} = I_{\mathrm{Audio}} + I_{\mathrm{Ripple}}$. Thus, the resistive losses of P_{FILT} will be treated in this work as a lower bound for the losses occurring at audio frequency. Theoretically, the resistive losses occur at both the audio frequency as well as on the switching frequency and its harmonics, but this is not included in this model.

The power loss model developed in this work is based on the power loss model in (2.14). However, since the simulation model has limitations P_{SW}, given in (2.21) will be replaced with P_{g} in (2.22). Additionally, due to limitations in accurately simulating the body-diode power losses P_{BD} (2.25), these losses are excluded in the analysis in this work. This exclusion introduces an unavoidable potential source of error, the adjusted power loss $P_{\mathrm{loss,adj}}$ can be expressed as:

$$P_{\mathrm{loss,adj}} = P_{\mathrm{Q}} + P_{\mathrm{CL}} + P_{\mathrm{g}} + P_{\mathrm{FILT}} \tag{2.27}$$

3

The Digital Class D Audio Power Amplifier

This chapter presents a detailed overview of the key components of a Class D audio power amplifier. It begins with a brief comparison of different amplifier classes and then gives an explanation regarding Class D amplifier operation, with an emphasis on the modulation stage. The chapter then details the characteristics of MOSFETs, covers information on various configurations of output stages, and discusses the output series resonance filter.

3.1 Overview of Audio Power Amplifier Classes

The audio power amplifier is an essential component in most audio systems. It is usually situated in the final stage of an audio amplifier block, where it transforms a low-power signal into a high-power output signal capable of driving a load. Traditional analog power amplifiers are classified into different classes depending on how they function. Some of these include Class A, B, AB, and others [23].

The class A amplifier is a commonly used amplifier topology that uses only one transistor in the output stage and has advantages such as high gain, low signal distortion, and good linearity [24, 25]. One drawback with this amplifier is that the transistor conducts current throughout the entire waveform cycle. This implies that the transistor is always on, which generates excessive heat and a continuous loss of power as given by (2.7). As a result, the amplifier has low power efficiency, with the theoretical maximum power efficiency of class A amplifiers only around 50% for a resistive load.

Class B amplifiers address some of the efficiency and heating problems associated with Class A designs and are essentially an improved version of the Class A amplifier. A generic Class B amplifier circuit uses a push-pull output configuration with two complementary transistors, where each transistor takes turns conducting each positive or negative cycle. Class B amplifiers can achieve higher power efficiency compared to Class A, with a maximum theoretical power efficiency of 78.5%, as given by (2.7). One major drawback associated with Class B amplifiers is that they suffer from linearity issues, where crossover distortion is the main limitation [25].

To eliminate the distortion issues associated with Class B amplifiers, the Class AB amplifier was developed [25]. It combines the topology characteristics of both Class A and Class B. The Class AB amplifier thus removes the crossover distortion of a Class B amplifier while offering higher efficiency than the Class A. The typical power efficiency for a Class AB amplifier falls in the range of 50% to 78.5%, as given by (2.7).

Class D amplifiers, which utilize high-speed switching transistors, were first introduced in the late 1950s, and are now regarded as one of the most promising technologies for audio power applications [17, 23]. They have a theoretical power efficiency of 100%, but in reality, it is between 90% and 95% for a resistive load, as calculated using (2.7) [26]. The development of these amplifiers has been driven by the need to fit more power capability into a smaller space while at the same time dissipating less heat. Something that is very attractive in various modern audio applications such as mobile phones and hearing aids. They have many advantages over previously named power amplifiers; these include small size, low heat dissipation, low power consumption, and high efficiency. One major drawback of Class D amplifiers has historically been the distortion associated with their switching operation, but with advancements in technology, this issue has become significantly less problematic.

A Class D amplifier can be configured as either a half or full bridge amplifier, both with their own advantages and disadvantages. The full bridge stage includes twice as many switching devices compared to the half bridge, resulting in higher costs and higher switching losses. However, the full bridge only requires half the power supply voltage to realize the same output voltage capability. Additionally, no current flows through the load when both sides are either off or on simultaneously, which can be utilized when optimizing switching. The full-bridge is also referred to as an H-bridge because the circuit has the shape of the letter H. An illustration of a H-bridge is shown in Fig. 3.1.

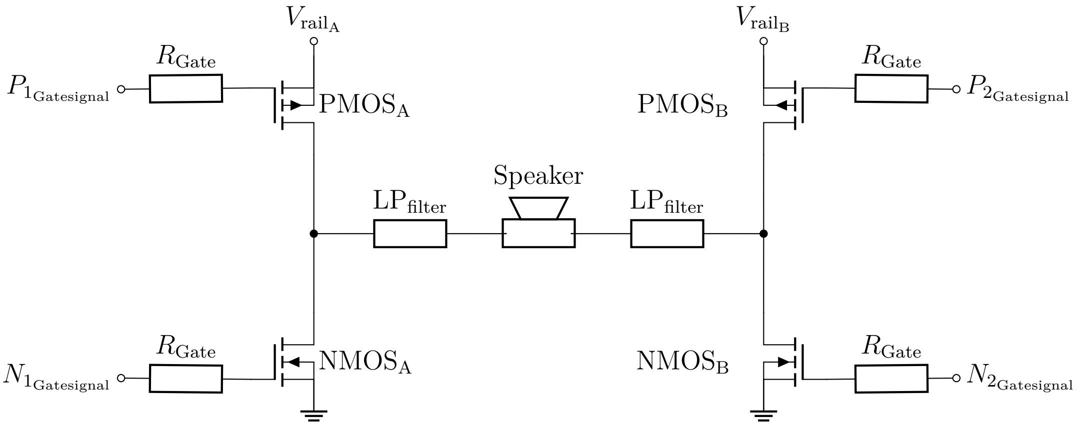


Figure 3.1: Schematic of an H-bridge Class D amplifier with high-side PMOS and low-side NMOS, driving a speaker through a low-pass filter.

In the H-bridge schematic shown in Fig. 3.1, N-channel MOSFETs are situated on the low-side, while P-channel MOSFETs are on the high-side. The gain of the Class D amplifier is equal to the ratio of the rail voltage V_{Rail}, and the input peak voltage $\hat{V}_{\text{i}}$:

$$\text{Gain} = \frac{V_{\text{Rail}}}{\hat{V}_{\text{i}}}. \tag{3.1}$$

3.2 Modulation Stage

The Class D amplifier modulator transforms the input audio signal into a high-frequency switching signal [27]. This signal is used to drive the switching stage on and off, thus amplifying the signal and delivering it to a load, either a loudspeaker or a transducer. The most widely used modulation techniques include pulse width modulation (PWM), Delta-Sigma ($\Delta\Sigma$) modulation, and self-oscillating control [23]. In this project, $\Delta\Sigma$ modulation will be utilized in combination with digital PWM.

3.2.1 PWM Signal Generation

PWM can be generated in both the digital and analog domains. In the analog domain, an audio frequency signal with a frequency typically ranging from 20 Hz to 20 kHz is compared to a high frequency carrier in the form of a sawtooth or triangular waveform with a frequency typically ranging from 0.1 to 1 MHz, using a comparator [27]. The output of the comparator is a high-frequency switching waveform in which the amplitude of the input signal is represented in the pulse width. This allows the input to control the duty cycle of the square wave, that is, the proportion of time the wave is in the high or low state.

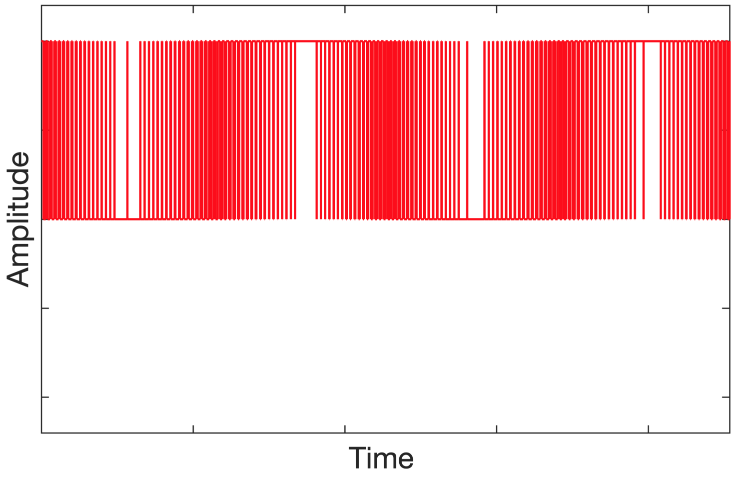
PWM is generally classified into two different variants, where the first is based on the edge characteristics of the carrier wave which can either be single-sided or double-sided. The second variant is determined by the number of output levels in the PWM signal, two (AD) or three (BD) [28]. A single-sided PWM signal employs a sawtooth carrier wave, whereas a double sided PWM signal uses a triangular carrier wave. These classifications are summarized in Table 3.1 [28].

Edge Detected	Output Levels
Single-Sided	Two (AD)
Single-Sided	Three (BD)
Double-Sided	Two (AD)
Double-Sided	Three (BD)

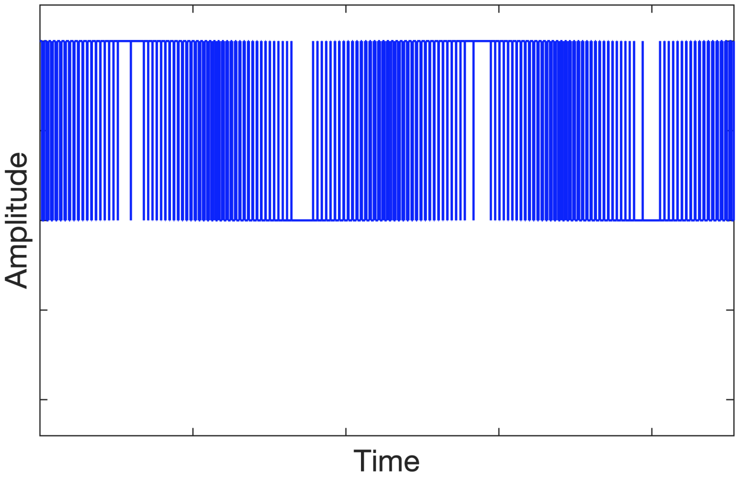
Table 3.1: Different PWM modulation variants based on edge detection and output levels.

Single-sided BD PWM, as highlighted in blue in Table 3.1 was used for the modulation stage in this thesis, due to the simplicity of integrating it into a simulation environment. The generation of the single-sided BD PWM is implemented in a DSP system with an internal timer, which counts up to a fixed value within one cycle. During each cycle the value of the timer is compared to the value of the input signals which ultimately defines the duty cycle of the generated PWM signal. For a single-sided BD PWM, the system has two input and output signals, with one being an inverted version of the other, representing the PWM fed to the A and B sides of the H-bridge. For simplicity, the input signals are usually simulated as sinusoidal waveforms.

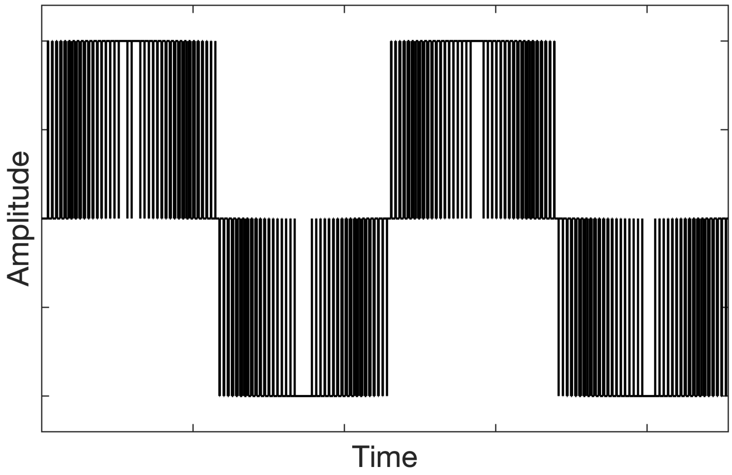
The resulting single-sided BD PWM signal is the difference between the PWM signals at A and B-side. In the H-bridge schematic shown in Fig. 3.1 the A side corresponds to the left side, while the B side corresponds to the right side. The PWM signal at A-side is illustrated in Fig. 3.2a and generated from the original input, and the PWM signal at B-side, illustrated in Fig. 3.2b is generated by the inverted input. The resulting signal is illustrated in Fig. 3.2c. The B-side should always receive an inverted version of the signal fed to A-side, regardless of the input signal.



(a) A-side of the BD modulated PWM.



(b) B-side of the BD modulated PWM.



(c) The resulting BD modulated PWM.

Figure 3.2: (a) A-side and, (b) B-side PWM signal, and (c) the resulting BD modulated output.

3.3 Class D Power Amplifier Stage

The proposed power stage of the amplifier consists of two circuits: the amplification stage and a series-resonant RLC circuit, which together amplify the signal and remove unwanted frequency components. The power stage in Class D amplifiers comprises power MOSFETs due to their high switching speed and the ability to easily achieve a rail-to-rail output using a relatively low input current [17].

In switching applications where MOSFETs are utilized, the aim is to rapidly transition between the on and off states of the transistor [29]. When the MOSFET is in the on state, it has low resistance, thus enabling maximum current flow with low impedance. In contrast, during the off state, the MOSFET presents high resistance, effectively stopping current flow. In high-frequency applications, the parasitic characteristics of the MOSFET are very important as they limit the switching speed.

The total gate charge value of a MOSFET Q_g is the amount of charge applied at the gate capacitance C_g required for the MOSFET to switch on [30]. Measured in Coulombs (C), this value typically falls within the nano Coulomb nC range. In high-frequency applications, minimizing the gate charge is of high importance to enhance efficiency. The amount of gate current I_g required from the drive circuit to switch the MOSFET in a desired time can easily be calculated with the following formulas, where V_GS once again denotes the gate to source voltage and t_SW is the switching time of the MOSFET:

$$Q_\mathrm{g} = C_\mathrm{g}\, V_\mathrm{GS}$$

and

$$I_\mathrm{g} = C_\mathrm{g} \frac{\mathrm{d}V_\mathrm{GS}}{\mathrm{d}t_\mathrm{SW}} \tag{3.2}$$

where

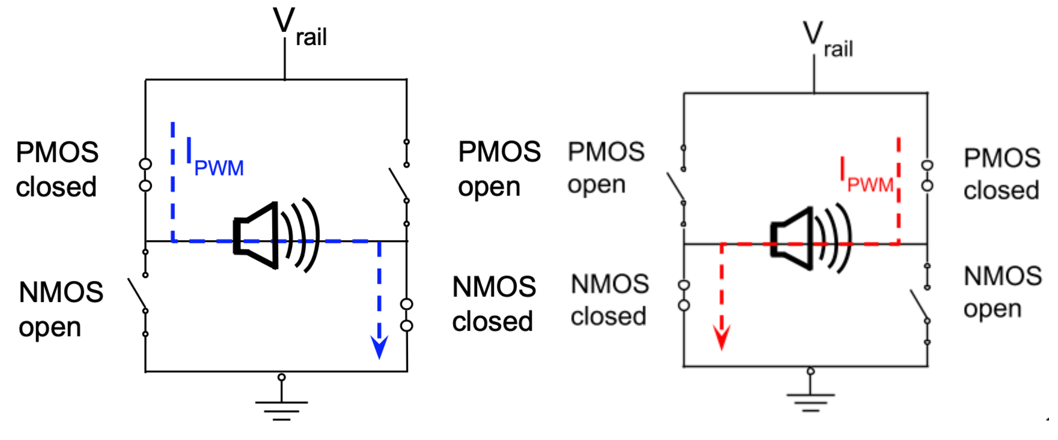
$$Q_\mathrm{g} = t_\mathrm{SW}\, I_\mathrm{g} \tag{3.3}$$

For example, a MOSFET with a gate charge of $Q_\mathrm{g} = 10$ nC can be switched on in $10\,\mu$s if a gate current of 1 mA, is applied to the gate or in 10 ns if the gate current is raised to 1 A. The role of the gate driver circuit is to amplify the output signal, thus enhancing the switching speed of the output stage.

3.3.1 Dead Time Control

MOSFETs operate like switches that, when turned on at the right moment, can amplify the input signal at the gate. There are several configurations that can be used to switch the MOSFETs depending on specific application. Fig. 3.3 illustrates the positive and negative transition modes, which lead to either a positive or a negative output. The positive transition mode is illustrated in Fig. 3.3a and the negative transition mode in Fig. 3.3b.

Furthermore, Fig. 3.4 presents the zero transition modes for an H-bridge configuration, all of which result in zero output. Fig. 3.4a shows the first zero transition



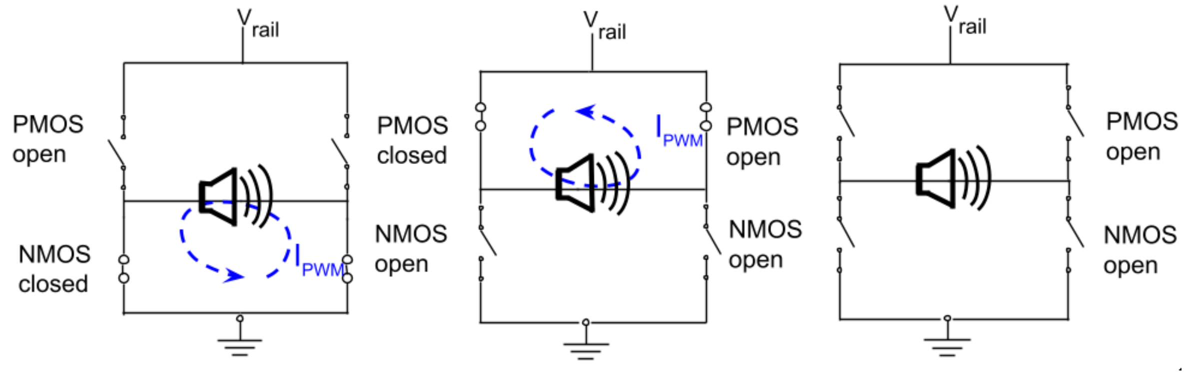
(a)

(b)

Figure 3.3: **(a)**First transition mode of the switches producing a positive output, and **(b)** second transition mode which gives a negative output.

mode where both high-side MOSFETs are open, Fig. 3.4b shows the second zero transition mode where both of the low-side MOSFETs are open and Fig. 3.4c illustrates the third zero transition mode where all MOSFETs are open.

When a PWM gate signal is applied to an H-bridge circuit utilizing MOSFETs, there can be a brief moment when both the high-side and low-side MOSFETs are turned on simultaneously or when one MOSFET turns on before the other has fully switched off. This can lead to a short circuit between the supply and ground and a large shoot-through current. Fig. 3.5 depicts the two possible short circuit scenarios, Fig. 3.5a illustrates a short circuit on the left side, and Fig. 3.5b illustrates a short circuit on the right side. To avoid this, dead-time control must be implemented in H-bridge circuits. Dead time is a transition period during which none of the MOSFETs on the same side of the H-bridge conduct at the same time. It can be controlled using analog circuitry or digitally with a microcontroller. Dead time also introduces significant distortion and must therefore be minimized while safely avoiding shoot-through [31, 17]. The dead time introduces a diode voltage drop as well as a voltage



(a)

(b)

(c)

Figure 3.4: Different switching modes that generate no output, therefore zero-modes.

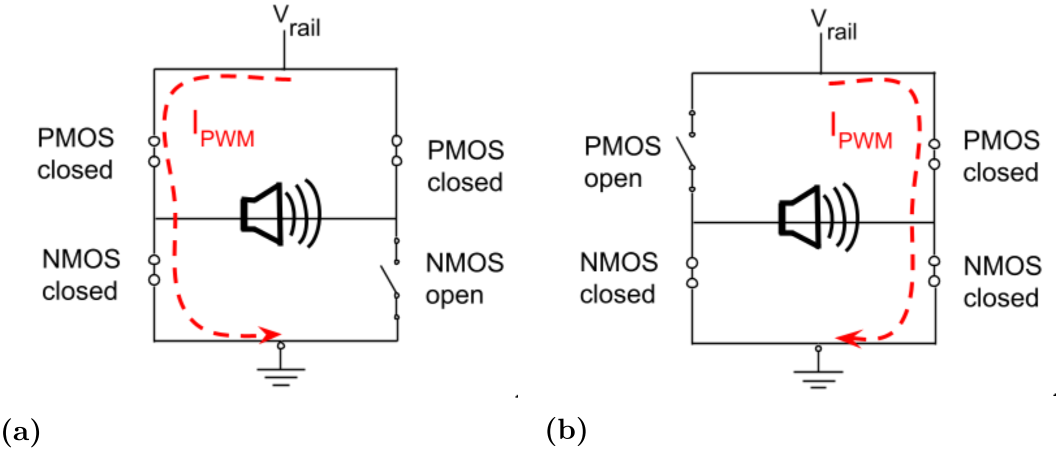


Figure 3.5: Unwanted switching modes that cause short circuits, which dead time control is designed to prevent.

increase in the circuit, proportional to the dead time. A rule of thumb is to have a dead time longer than the switching time of the slowest MOSFET.

3.3.2 Series-Resonant Circuits

Fig. 3.6 shows an illustration of a series resonance circuit. The inductive reactance of an inductor X_L rises linearly with the frequency, meaning it is proportional to the frequency ($X_\mathrm{L} \propto f$) and is illustrated as the red straight linear arrow in Fig. 3.7. The value of the reactance X_L is given by:

$$X_\mathrm{L} = 2\pi f L = \omega L$$

In contrast to the inductor, the capacitive reactance of a capacitor X_C is inversely proportional to the frequency ($X_\mathrm{C} \propto f^{-1}$) and is illustrated as the blue hyperbolic curve in Fig 3.7. The value of the reactance X_C is given by:

$$X_C = \frac{1}{2\pi f C} = \frac{1}{\omega C}$$

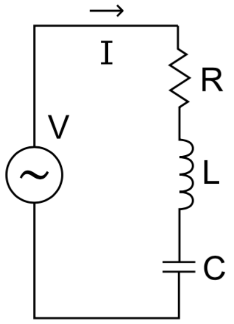


Figure 3.6: Series resonance circuit showing an RLC circuit consisting of an AC voltage source, a resistor, an inductor, and capacitor [32].

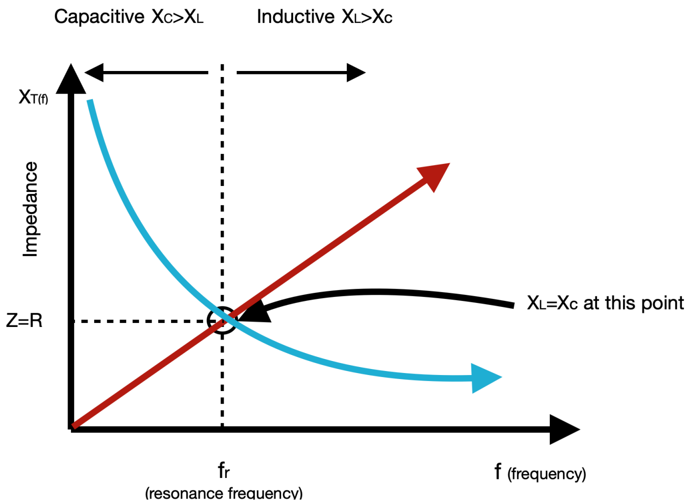


Figure 3.7: Impedance versus frequency for capacitance and inductance, and highlighting the resonance frequency.

By setting $X_\mathrm{L} = X_\mathrm{C}$, we have:

$$2\pi f L = \frac{1}{2\pi f C}$$

Rearranging terms:

$$f^2 = \frac{1}{(2\pi L)(2\pi C)} = \frac{1}{4\pi^2 LC}$$

Taking the square root:

$$f = \sqrt{\frac{1}{4\pi^2 LC}}$$

Thus, the resonant frequency is given by:

$$f_r = \frac{1}{2\pi\sqrt{LC}} \quad \text{(Hz)} \tag{3.4}$$

Alternatively, the angular resonant frequency is:

$$\omega_r = \frac{1}{\sqrt{LC}} \quad \text{(rads)}$$

Another important metric is the Quality or Q factor of the circuit, which essentially determines the sharpness of the resonance peak. The resistance value determines the width of the curve; a lower resistance results in a sharper and narrower curve. The Q-factor is given by [33]:

$$Q = \frac{1}{\omega_r RC} \tag{3.5}$$

4

Digital-to-Analog Converters

This chapter provides a theoretical introduction to digital-to-analog converters (DAC). The discussion begins with an overview of conventional Nyquist rate converters, followed by an introduction to oversampling converters and $\Delta\Sigma$ converters. The chapter will specifically highlight differences in the quantization noise processes for each type of converter and demonstrate why $\Delta\Sigma$ DACs are often preferred in converter systems utilizing oversampling.

4.1 Decibels Relative to Full Scale

The unit decibel relative to full scale, or dBFS, is essential in digital audio systems. It represents the input signal amplitude relative to the maximum achievable signal amplitude of a system before a phenomenon called clipping occurs. Clipping occurs when the system receives an audio input signal with an amplitude that exceeds the system's maximum representable level. In digital audio systems, 0 dBFS represents the highest possible level without distortion. Due to the high risk of clipping, real-world playback audio systems almost always operate at input amplitude levels below 0 dBFS. More common signal amplitude values fall within the range of -1 to -20 dBFS to minimize distortion. The unit dBFS is also used to quantify the dynamic range of a digital audio signal.

4.2 Nyquist-Rate Digital-to-Analog Converters

Digital to analog conversion (DAC) is a process that translates a discrete time digital input to a continuous time analog output signal [34]. An N-bit Nyquist-rate DAC operates at a sampling rate at or above the Nyquist criterion, $f_\mathrm{s} \geq 2f_\mathrm{b}$ where f_s is the sampling rate and f_b is the bandwidth of the sampled signal. Each sample is processed individually, and the converter lacks memory [35].

4.2.1 Quantization in a Nyquist DAC

The DAC's performance is highly dependent on the accuracy of its circuit components. In addition, when the application demands higher resolution and linearity, the converter suffers from its slow operating speed. For the DAC to accurately represent the input, the number of quantization levels as defined by 2^N should be maximized. The quantization step size represents the distance between two consecutive quantization levels and is defined as:

$$q = \frac{V_{\text{max}} - V_{\text{min}}}{2^N - 1} \tag{4.1}$$

where V_{max} and V_{min} are the maximum and minimum values of the input signal. Since digital signals are discrete, the conversion process always introduces quantization error, which is the difference between the quantized output and the input, defined by $e_{\text{q}} = Q(x) - x$, where Q(x) is the quantized output and x is the input. The quantization error is always constrained by the range $-\frac{q}{2} \leq e_{\text{q}} \leq \frac{q}{2}$, as can be seen in the probability density function (PDF) of the quantization error in Fig 4.1a.

The quantization operation is a nonlinear process, because of the mapping of continuous values to discrete levels [36, 35]. Due to this reason, three simplifying assumptions regarding the quantization error properties are generally made.

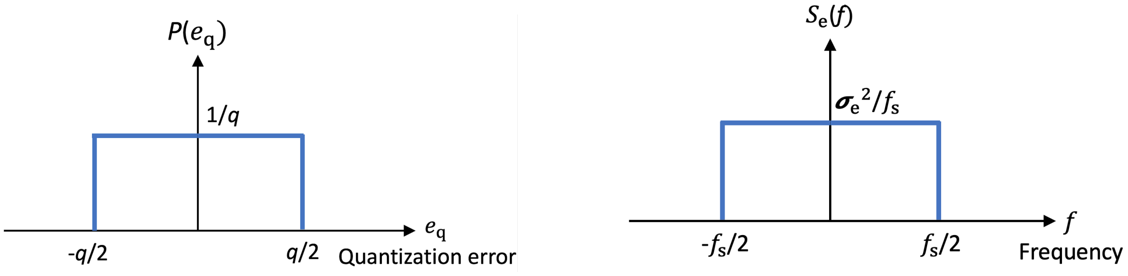
1. The error sequence, e_{q} is assumed to be a random process with stationary properties.
2. The error sequence, e_{q} is assumed to be independent across time and independent with the input sequence x.
3. The probability density function (PDF) of the quantization error sequence e_{q}, is assumed to be uniform over the range $-q/2 \leq e_{\text{q}} \leq q/2$.

Assumption 3 gives the following [36]:

$$p(e_{\text{q}}) = \begin{cases} \frac{1}{q}, & |e_{\text{q}}| \leq \frac{q}{2} \\ 0, & |e_{\text{q}}| > \frac{q}{2} \end{cases} \tag{4.2}$$

which can be visualized in Fig. 4.1a.

The following assumptions allow for the error to be visualized as white noise, thus the power spectral density (PSD) of the error will have uniform distribution within the Nyquist band as shown in 4.1b.



(a) PDF of the quantization error in an ideal Nyquist-rate DAC.

(b) PSD of the quantization error in an ideal Nyquist-rate DAC.

Figure 4.1: PDF and PSD of the quantization error in an ideal Nyquist-rate DAC.

With these assumptions, the formula for the average quantization noise can be expressed as:

$$\bar{e_{\text{q}}} = E\{e_{\text{q}}\} =$$

$$\int_{-\infty}^{\infty} e_{\text{q}}\, p(e_{\text{q}})\, de_{\text{q}} =$$

$$\frac{1}{q}\int_{-\frac{q}{2}}^{\frac{q}{2}} e_{\text{q}}\, de_{\text{q}} = 0$$

and the power of the quantization noise is represented by:

$$\sigma_e^2 = E\{(e_{\text{q}} - \bar{e_{\text{q}}})^2\} =$$

$$\int_{-\infty}^{\infty} e_{\text{q}}^2\, p(e_{\text{q}})\, de_{\text{q}} =$$

$$\frac{1}{q}\int_{-\frac{q}{2}}^{\frac{q}{2}} e_{\text{q}}^2\, de_{\text{q}} = \frac{q^2}{12} \tag{4.3}$$

by substituting (4.1) into (4.3), under the assumption that $V_{\text{max}} = 1$ and $V_{\text{min}} = -1$ the following expression for quantization noise power is obtained

$$\sigma_{\text{e}}^2 = \frac{q^2}{12} = \frac{1}{3(2^N - 1)^2} \simeq \frac{1}{3(2^{2N})} \tag{4.4}$$

4.2.2 Signal-to-Noise Ratio in a Nyquist DAC

To determine the SNR for a Nyquist DAC-generated PCM signal, the signal power must first be calculated. The input signal being quantized is assumed to be sinusoidal signal with amplitude A, expressed as: $x(t) = \text{A}\cos(\frac{2\pi t}{T})$.

The quantization error e_x is defined as the difference between the input signal $x(t)$ and its quantized version $\hat{x}(t)$.

$$e_x = \hat{x}(t) - x(t) \tag{4.5}$$

The average power of the signal is given by [36]:

$$\sigma_{\text{x}}^2 = \frac{A^2}{2} \tag{4.6}$$

By substituting the resulting terms for average noise power in (4.4) and the average signal power in (4.6) into the formula for SNR provided in (2.1), the following expression is obtained [36]:

$$\text{SNR(dB)} = 20log_{10}\frac{\sigma_{\text{x}}}{\sigma_{\text{e}}} = 10log_{10}\frac{3 \times 2^{2N}}{2} \simeq 20\log_{10} A + 6.02\, N + 1.76$$

This expression shows that the signal-to-noise ratio improves by around 6 dB for every increment in the number of bits in the quantizer. An audio signal quantized with 16-bits resolution has a theoretical maximum SNR of 98.08 dB.

4.3 Oversampling Converters

One way to ease the accuracy demand of analog devices is by using oversampling converters. An oversampling converter operates at a sample rate higher than the Nyquist criterion according to $OSR = \frac{f_\mathrm{s}}{2f_\mathrm{B}}$, where OSR abbreviates the oversampling ratio [35]. The OSR value is generally set to a multiple of 2, where the most common values are between 8 and 512. Oversampling converters that incorporate a loop filter with a feedback structure, such as the $\Delta\Sigma$ modulator, rely on preceding input values to generate the output, unlike Nyquist-rate converters, which process each sample individually. An oversampling converter requires more advanced digital circuitry and is slower than the Nyquist-rate converter, but it increases the accuracy and linearity.

4.3.1 $\Delta\Sigma$ Digital-to-Analog Converter

A common type of oversampling converter is the Delta Sigma ($\Delta\Sigma$) DAC. In comparison to other oversampling converters, it includes a loop filter, which subtracts the difference between the quantized output and the input signal, a process known as noise shaping. Noise shaping shifts the noise within the band of interest to higher frequencies outside of the band of interest; this concept will be further described in chapter 5. $\Delta\Sigma$- modulation is mainly used to convert signals from a low-rate, and large-word-length format to a high-rate and low-word-length format [35]. For example, a signal input of 16 bit and a f_s of 44.1 kHz, could be upsampled by 4, to a f_s of 176.4 kHz, and a 1 bit word length. The DAC can be fully implemented in the digital domain, and the modulation order and the OSR of the converter determine the performance. A general first order $\Delta\Sigma$ DAC is composed of two main blocks, the interpolation stage where the upsampling and filtering occur and the $\Delta\Sigma$ modulator, shown in Fig. 4.2. As its name suggests, the modulator consists of a negative feedback loop Δ, an accumulator stage called Σ and an N-bit quantizer.

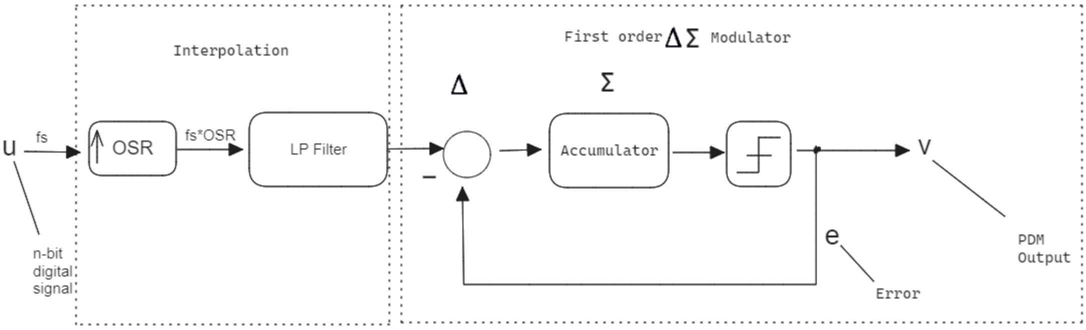


Figure 4.2: General block schematic of a first order ΔΣ DAC.

A comparison of the different shapes of quantization noise, for Nyquist-sampled, oversampled and noise-shaped signals is illustrated in Fig. 4.3, showing how noise-shaping shifts the noise within the band of interest to higher frequencies outside of the band of interest. The figure highlights the desirable attributes regarding noise shaping.

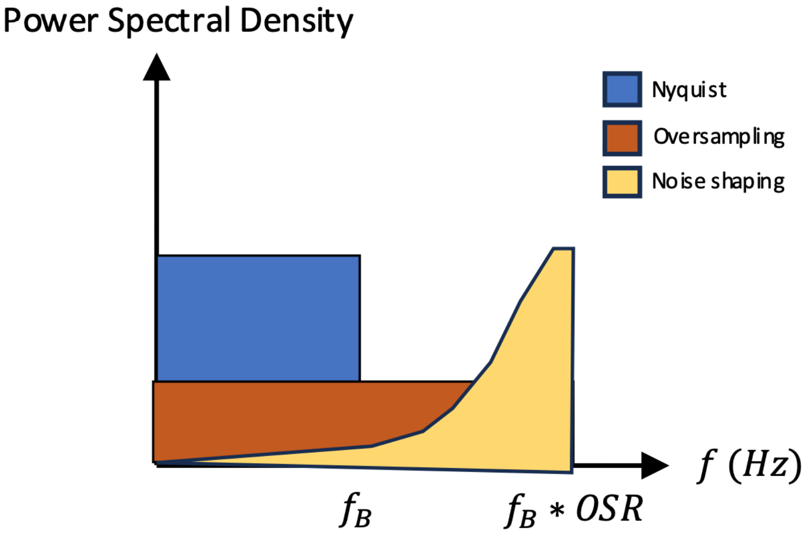


Figure 4.3: Comparison of the PSD of quantization noise for Nyquist sampling, oversampling, and noise-shaped signals.

5

Digital $\Delta\Sigma$ Modulation

In this section, the fundamental theory regarding digital $\Delta\Sigma$ modulation is presented. It begins with an introduction to the interpolation stage shown in Fig. 4.2, covering the principles and role of upsampling and filtering. Following this, the $\Delta\Sigma$ modulation loop shown in Fig. 4.2 is outlined, together with key principles such as noise shaping, and an overview of $\Delta\Sigma$ converter performance metrics.

5.1 Interpolation

Interpolation has the role of increasing the sampling rate of a digital input signal while removing or attenuating spectral replicas created during upsampling [35]. Interpolation is thus the combined function of upsampling and low pass filtering of an input signal.

5.1.1 Upsampling

Upsampling is the process of increasing the sample rate of a signal by creating additional data points from an existing set of data points [35]. A common method is zero stuffing, which is performed by inserting OSR-1 zeros between each sample. This increases the sampling frequency f_s to $f_\mathrm{s} \times OSR$. When a signal undergoes upsampling, it introduces spectral replicas at $f_\mathrm{s}, 2f_\mathrm{s}, ..., (OSR-1)f_\mathrm{s}$ where f_N is the Nyquist frequency. To remove unwanted spectral replicas, a digital low-pass filter must be applied to the upsampled signal.

5.1.2 Digital Filters

A digital filter is a system that performs a mathematical operation on a discrete time sampled input to modify the input signal [34, 37]. The general transfer function for a digital filter is:

$$H(z) = \frac{b_0 + b_1 z^{-1} + ...b_n z^{-n}}{1 + a_1 z^{-1} + ...a_n z^{-n}} = \frac{Y(z)}{X(z)}$$

where X(z) represents the z-transform of the input signal and Y(z) represents the z-transform of the output signal. Taking the inverse z-transform yields the difference equation

$$y_n = -a_1 y_{n-1} - a_2 y_{n-2} - ... + b_0 x_n + b_1 x_{n-1} + b_2 x_{n-2} + ... \tag{5.1}$$

which is the difference equation of the filter, and where a_n and b_n are the filter coefficients [34]. The filter coefficients can be calculated to shape and filter the input signal in a certain way. The most common digital filters are infinite impulse response (IIR) filters and finite impulse response (FIR) filters [37]. IIR filters rely both on input and earlier output samples, as described in (5.1). They are usually of low order and requires less filter coefficients than FIR filters, which makes them faster. But IIR filters are not always stable and the phase shift is not linear. In comparison, FIR filters do not rely on prior output samples and can be described by putting all the a_i coefficients to zero in (5.1). This filter is slower but has inherent stability and a linear phase. Due to this fact, FIR filters are more suitable for audio applications, as the phase information is more important than the speed [38].

5.1.3 Half-Band FIR Filter

A filter suitable for interpolation by a factor 2 are half-band filters [39]. The frequency response of a half-band filter should satisfy $H(e^{j(2\pi\frac{f}{f_s})} + H(e^{j2\pi(0.5+\frac{f}{f_s})}) = 1$, which causes nearly half of the impulse response samples to be zero, and therefore no computation is required for half of the taps in the filter. The ideal half-band FIR filter response is visualized in Fig. 5.1 represented by the black curve, with the light blue ideal magnitude scale on the left y-axis. The blue, red, and green curves correspond to filters with 51, 15, and 11 taps, respectively with the orange magnitude scale on the right y-axis. The cascaded configuration of these filters is visualized in Fig. 5.2.

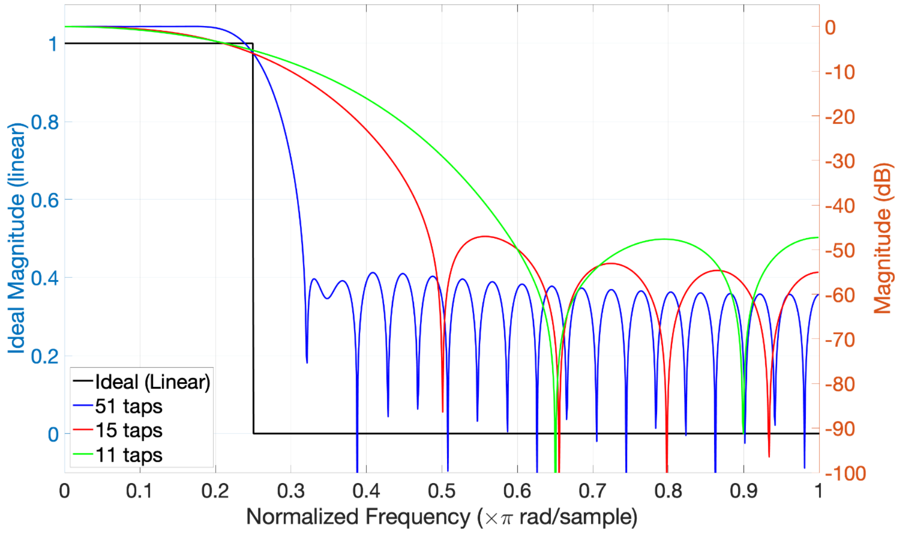


Figure 5.1: Frequency response of an ideal half-band FIR filter compared with compared with practical implementations using 51, 31, and 15 taps.

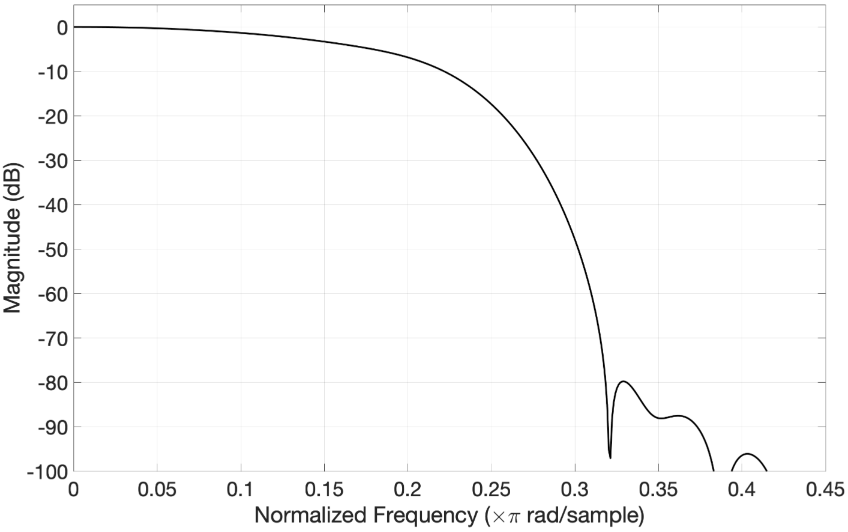


Figure 5.2: Cascaded configuration of the implemented half-band FIR filter with 51, 31 and 15 taps.

Fig. 5.3 depicts a signal upsampled by 2, stuffing 2-1 zeros in between every sample and creating a spectral replica at 9000 Hz. As can be seen above, by putting this through a half-band FIR filter, the signal is averaged and the spectral replica is removed. The original signal has a frequency of 1000 Hz, and a f_s of 10 kHz. After upsampling, the sampling frequency is $F_s = 20$ kHz.

To achieve a high oversampling ratio using only half-band filters, multistage interpolation must be employed [39]. In other words, cascaded upsampling by two combined with complementary filtering stages. For every interpolation stage, the demand on the transition band can be relaxed.

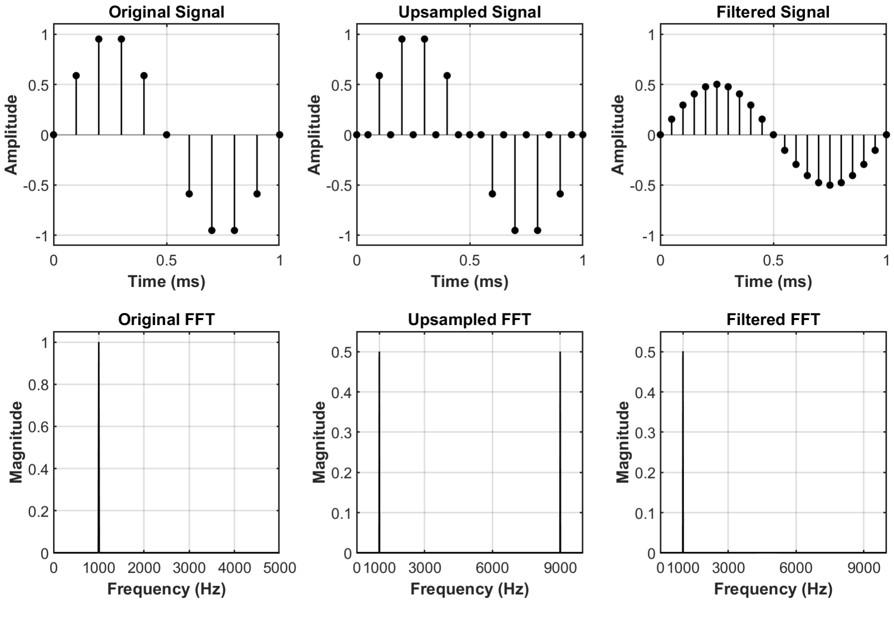


Figure 5.3: Sinusoidal signal upsampled by a factor of 2 and filtered by a FIR filter.

5.2 First Order ΔΣ Modulator Feedback Loop

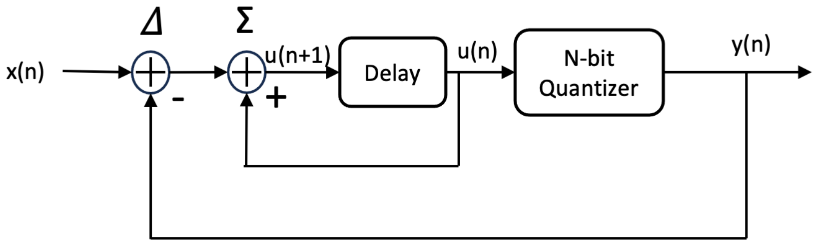
As discussed in section 4.3.1, the first-order ΔΣ modulator feedback loop generally consists of three components: a feedback difference operation Δ, an accumulator Σ and an N-bit quantizer. Fig. 5.4 illustrates two block diagrams of a first-order ΔΣ modulator, Fig 5.4a presents the time-domain representation, and Fig. 5.4b depicts the corresponding z-transform block.

The N-bit quantizer in the ΔΣ modulation loop introduces a quantization error described by $e = u - y$, where u is the input to the quantizer and y is the output. This quantization error is assumed to be uniformly distributed, making the model linear and easy to analyze in the Z-domain [35].

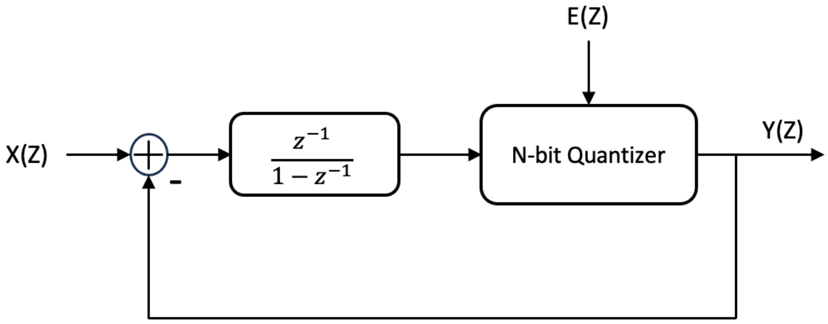
The output of the Z-block system in Fig. 5.4b can be described using the signal transfer function (STF), and the noise transfer function (NTF) as:

$$Y(z) = STF(z)X(z) + NTF(z)E(z), \tag{5.2}$$

where $Y(z)$ is the output, $X(z)$ is the input and $E(z)$ is the quantization error of the system.



(a) Time-domain block diagram of a first-order ΔΣ modulator with an N-bit quantizer.



(b) Z-transform block diagram of a first-order ΔΣ modulator with an N-bit quantizer.

Figure 5.4: Block diagrams of a first-order ΔΣ modulator: (a) time-domain representation and (b) corresponding z-transform representation.

Since the accumulator in the Z-block diagram in the system in Fig. 5.4b has the transfer function:

$$H(z) = \frac{z^{-1}}{1 - z^{-1}}, \tag{5.3}$$

the STF can thus be written as:

$$STF(z) = \frac{Y(z)}{X(z)} = \frac{H(z)}{1 + H(z)} = z^{-1}, \tag{5.4}$$

which is essentially just a unit delay.

And the NTF can be written:

$$NTF(z) = \frac{Y(z)}{E(z)} = \frac{1}{1 + H(z)} = 1 - z^{-1}, \tag{5.5}$$

which shows that the NTF is essentially a first order high pass transfer function, and is the part that contributes to the noise shaping in the ΔΣ modulator.

5.3 Higher Order ΔΣ Modulator Feedback Loops

A ΔΣ modulation loop can be of different order and employ different feedback structures. The number of integrators and feedback paths present in the modulation loop determines the order. Fig. 5.5 shows the Z-transform block of the second-order ΔΣ modulator.

The second-order modulator's transfer function in the Z-domain is given by:

$$Y(z) = X(z)z^{-1} + E(z)(1 - z^{-1})^2 \tag{5.6}$$

from this expression, the NTF can be formulated as [36]:

$$NTF(f) = (1 - e^{-j2\pi \frac{f}{f_s}})^2, \tag{5.7}$$

and furthermore the in-band noise power can be expressed as:

$$\sigma_n^2 = \frac{\sigma_e^2 \pi^4}{5 \times OSR^5} \tag{5.8}$$

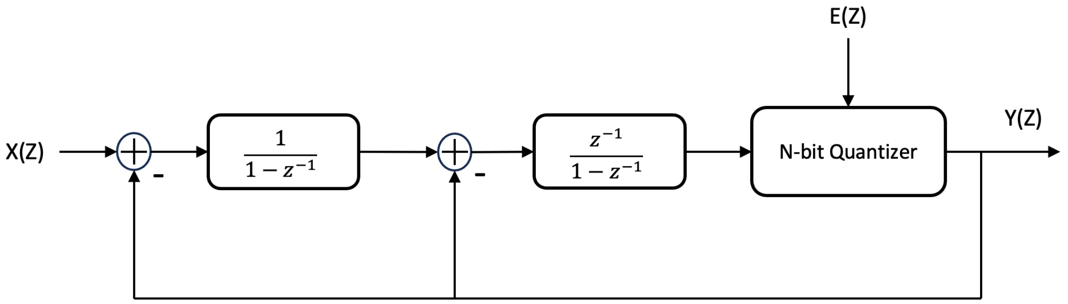


Figure 5.5: Block diagram of the z-transform of a second order ΔΣ modulator.

ΔΣ modulators of first and second order are inherently stable and no filter is thus needed in the feedback loop. For higher order modulators there are known stability issues, so the filters need to be designed carefully.

The maximum SNR for a second order ΔΣ modulator can be approximated by [36]:

$$\mathrm{SNR(dB)} \approx 20\log_{10}A + 6.02N + 1.76 + 10\log_{10}(5) - 9.94 \times 2 + 3.01(2 \times 2 + 1)r, \quad (5.9)$$

where N is the number of bits in the quantizer, and r denotes the exponent of the OSR, defined as:

$$r = \log_2(OSR). \quad (5.10)$$

For example, an OSR of 8 corresponds to 2^3, so $r = 3$ and OSR $= 2^r$.

6

Method

This chapter explains the methodology used in this project and gives a description of the design and simulation of each part. It starts with a block diagram of the proposed Class D power amplifier system, shown in Fig. 6.1. The entire system is developed within a simulation environment, with the intention of reflecting the practical system. To replicate a real-world scenario as closely as possible, and align with the configurations of a 16-bit 44.1 kHz digital audio sound card using the Integrated Inter-IC Sound Bus (I^2S) format, the simulated input signal is designed with the same bit depth and sampling frequency.

The input signal then goes into the DSP stage, where it goes through a series of upsampling and half-band filtering, thus increasing the sampling frequency f_s with $OSR = 8$ to $F_s = f_s \times 8$, to align the $\Delta\Sigma$ modulated output frequency with the BD PWM switching frequency $F_s = 352.8\,\text{kHz}$.

After the interpolation stage the signal with sampling frequency F_s enters the $\Delta\Sigma$ modulator which reduces the resolution of the signal down to 7-bits. The output of the $\Delta\Sigma$ modulator is then converted into a BD PWM signal with dead-time control that is fed into the H-bridge power stage in LTspice software, which drives the piezoelectric actuator load.

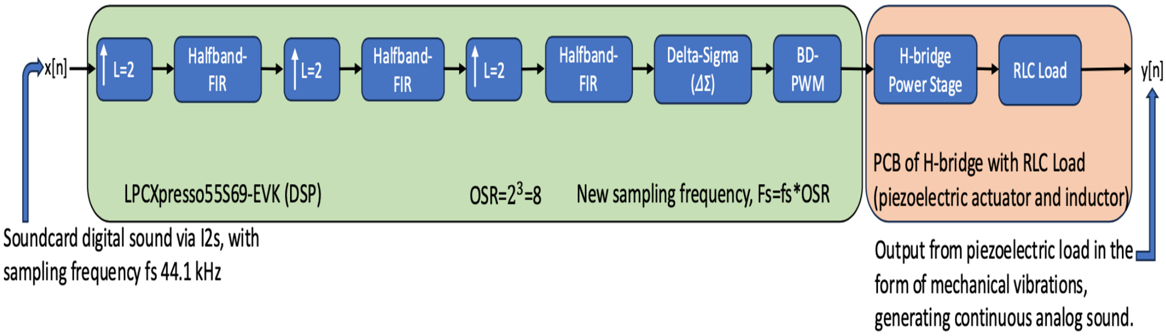


Figure 6.1: Block diagram of the proposed system, with the DSP section highlighted in green and the circuit section in peach.

6.1 Selection of Microcontroller

An extensive literature review was conducted in order to select a suitable microcontroller (MCU) for this work. The objective was to find a microcontroller that met specific criteria: a high core frequency to facilitate dead-time control, a core type M33, M0+, or M4 with 32-bit operation, Phase-Locked Loop (PLL), and the availability of a development kit (Devkit). A development kit is required as it makes it easier to use the microcontroller and all its functions. A PLL can be used to control the frequency in the microcontroller, between the input output and the internal clock. An ARM processor was required since it would make it easier to integrate with Cochlear's products. Table 6.1 presents a non ranked list of the top seven identified candidates through this process, where the two chosen candidates MAX32650 and LPCXpresso55S69 are lined in blue, respective green. The LPCXpresso55S69 was the main micrcontroller for this work and was used together with its development kit, the LPCXpresso55S69-EVK.

6.2 Design and Simulation in MATLAB

Initially, simulations of the second order $\Delta\Sigma$ modulator and the BD PWM will be on the ideal non-quantized $\Delta\Sigma$ modulation version with a floating-point sinusoidal input, to evaluate its performance. Later, a 16-bit quantized sinusoidal input will be introduced to reflect real-world scenarios.

6.2.1 Interpolation

Interpolation was implemented using built-in upsampling functions to increase the sample rate, along with a custom made half-band FIR filter. The built-in "filtfilt" function was used to filter with a zero-phase delay. The half-band FIR filter was used due to its simplicity and efficiency when dealing with signals that use zero-stuffing and that are upsampled by a factor of 2. A second order $\Delta\Sigma$ modulator was implemented in MATLAB using an accumulator-based structure as proposed by [35]. The second order modulator was chosen for its simplicity in implementation and its inherent stability as its poles lie within the unit circle [40].

Table 6.1: Non-ranked list of the top 7 microcontroller candidates

Model	*Core*	*Interfaces*	*PLL*	*Devkit*
Apollo3 Blue Datasheet	M33, 150 MHz	I2C or I2S	No	Yes
LPC55S6x	M33, 150MHz	SPI, I2C, or I2S	Yes	Yes
Renesas RE01	M0+, 64 MHz	I2C or simple SPI	No	Yes
MAX32680	M4, 120 MHz	Two I2C and I2S	Yes	Yes
MAX32666	M4, 96 MHz	I2C, QSPI or I2S	Yes	Yes
EFR32BG22	M33, 76.8 MHz	I2S or SPI	Yes	Yes
MAX32650–MAX32652	M4, 120 MHz	SPI, I2C or I2S	No	Yes

To reduce spectral leakage, the fundamental frequency f_0 of the input signal sampled at $f_\mathrm{s} = 44.1$ kHz, is generated using the following formula:

$$f_0 = \frac{f_\mathrm{s}}{Q}$$

where Q is an integer determined by the equation:

$$n_\mathrm{samples} = P \times Q$$

where P is the number of periods. For example, by using a Q value of 10 and a P value of 1000, a signal with the fundamental frequency $f_0 = 4410$ Hz is generated. The signal is then interpolated to an OSR of 8, corresponding to a sampling frequency F_s of 352.8 kHz.

6.2.2 Design and Implementation of the ΔΣ modulator

An important metric for the modulators performance is the SNR and SINAD, the evaluation of which is done in the frequency domain [41]. By summing the frequency bins in the band of interest and comparing them with the overall frequency band, the SINAD can be calculated according to (2.5). When calculating the SNR, the power of the harmonics was also excluded, according to (2.1). This analysis was performed on the output of the ΔΣ modulator as well as for the BD PWM signal. Fig. 6.2 depicts the half-sided power spectrum, where the blue curve represents the power spectrum, the yellow lines are the noise bins and the red lines are the signal bins of the fundamental tone. The noise bins and the signal bins are summed independently of each other and compared, to determine the SNR.

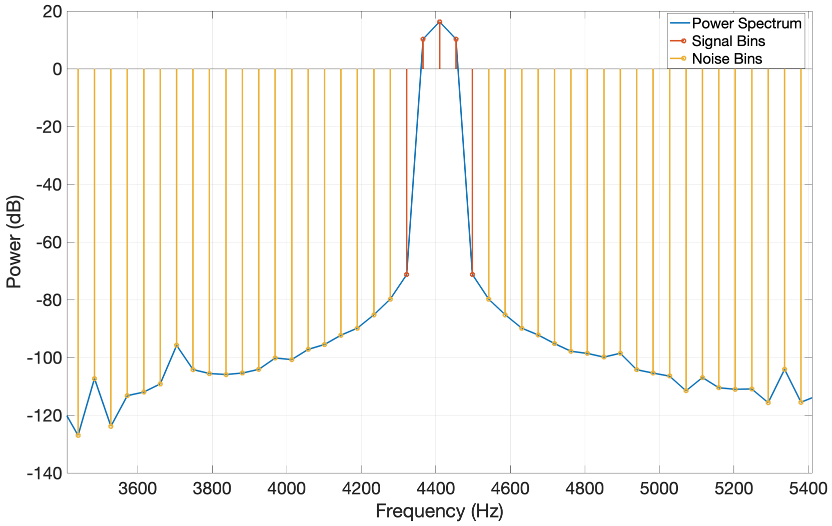


Figure 6.2: Power spectrum of the ΔΣ modulated output showing the signal components and noise components, used for SNR analysis.

Frequency analysis experiments were initially conducted with different quantizer bit resolutions of the second order ΔΣ modulator to evaluate its performance. The PSD of the ideal second order ΔΣ modulator with a non-quantized input, and with output bit resolutions ranging from 7 to 3 bits, is shown in Fig. 6.3. It clearly demonstrates that as the number of bits decreases, the noise floor shifts upward reflecting the increase of quantization noise. Based on this observation, together with the aim of maximizing the performance a 7-bit quantizer in the ΔΣ modulator was chosen.

In Fig. 6.4 the PSD of the second order 7-bit ΔΣ modulator with and without 16-bit quantized input is shown. The blue curve represents the PSD of the ideal ΔΣ modulator, while the orange curve corresponds to the PSD of the ΔΣ modulator with 16-bit quantized input. The noise floor is slightly lower and the power of the harmonic components are slightly reduced in the ideal ΔΣ modulator.

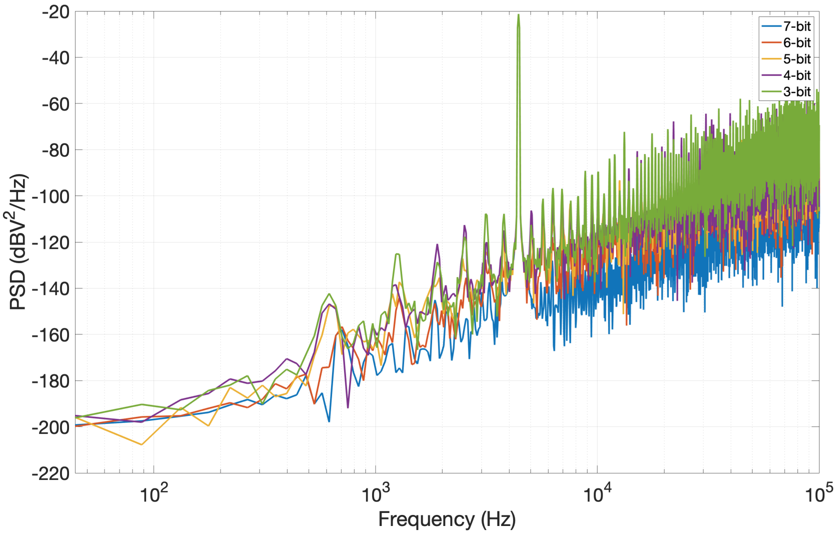


Figure 6.3: PSD of an ideal second order ΔΣ modulator driven by a non-quantized sinusoidal input with f_0=4410 Hz, with output resolution ranging from 7 to 3 bits.

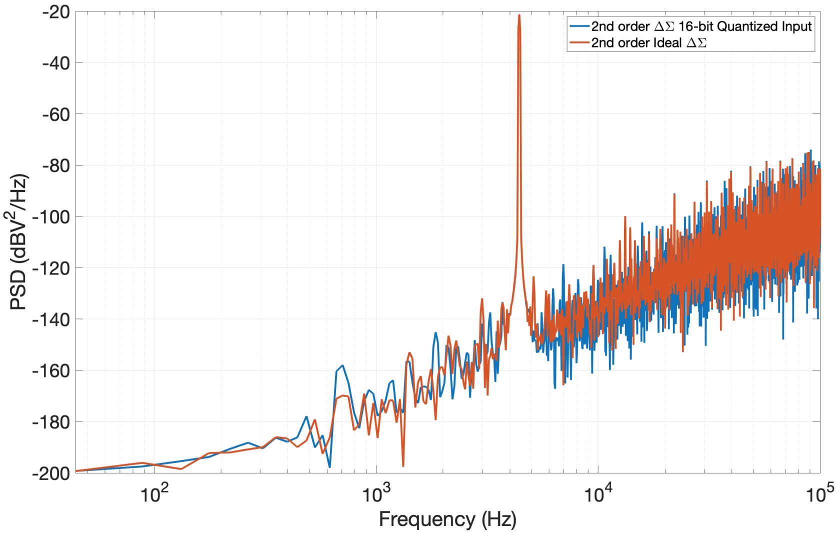


Figure 6.4: PSD of a second order 7-bit $\Delta\Sigma$ modulator driven by both a non-quantized and a 16-bit quantized sinusoidal input.

6.2.3 Conversion of $\Delta\Sigma$ to BD PWM

To convert the $\Delta\Sigma$ modulated signal into BD PWM, the principles of Single-Sided BD modulation, as explained in Section 3.2.1, were employed. Initially, this was implemented without dead-time control, working with a counter of 256 values, with the intention to mimic the counter function in the microcontroller. This results in a required clock frequency, denoted by f_{PWM}. The required clock frequency controlling the BD PWM is given by $f_{\text{PWM}} = 256 \times F_s = 256 \times 352.8\,\text{kHz} = 90.3168\,\text{MHz}$.

Since the counter has a value of 256, the output signals of the $\Delta\Sigma$ modulator lies in the interval [-128 to 127]. The output signals of the $\Delta\Sigma$ modulator were then scaled to the range [0 to 256], where a value of 0 indicates that the BD PWM signal remains low for the entire period, and 256 indicates it stays high for the entire period. For each output signal sample, the duty cycle value determines the proportion of the BD PWM period that the signal stayed high or low. The BD modulated PWM signal together with the $\Delta\Sigma$ modulated output is shown in Fig. 6.5. In this plot the $\Delta\Sigma$ modulated output is scaled to match the amplitude of the BD PWM. It is evident that the duty cycle of the pulse widths in the BD PWM signal accurately represents the amplitude values of the $\Delta\Sigma$ modulated output signal.

Fig. 6.6 presents the PSD of the BD modulated PWM signal driven by both the ideal version with a non-quantized and a 16 bit quantized sinusoidal input. The curve in blue represents the PSD of the BD PWM with a 16 bit quantized input and the orange curve represents the PSD of the ideal version. The harmonics of the fundamental sinusoidal f_0 can be observed at regular intervals $n = 1, 2, 3.... \times f_0$. In the greater frequencies, the PWM switching harmonics frequencies are also easily distinguished, with harmonics appearing at $n = 1, 2, 3.... \times f_{\text{PWM}}$, with upper and lower sidebands at $f_{\text{PWM}} \pm n \times f_0$.

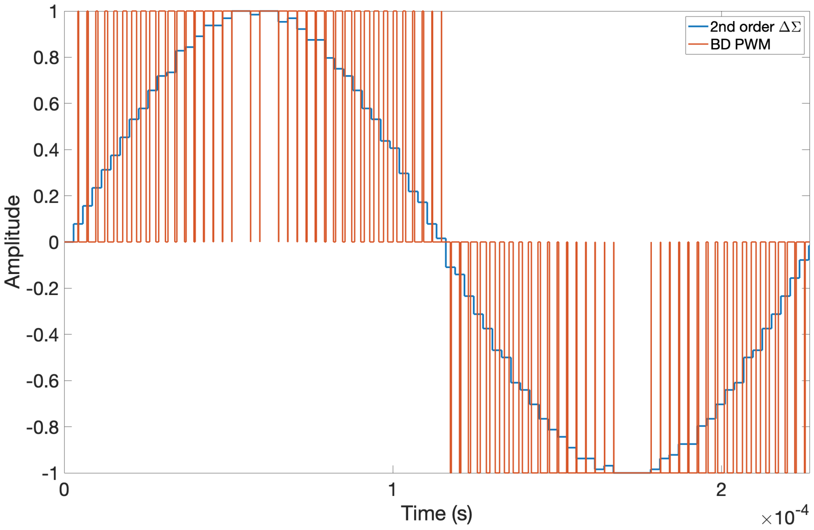


Figure 6.5: One period of the scaled $\Delta\Sigma$ modulated output compared with the BD PWM signal.

6.2.4 Performance Analysis of BD PWM

Other key performance metrics evaluated were the THD and THD+N values of the BD PWM signal, calculated using (2.2) and (2.3), respectively. These metrics were analyzed using the different tests illustrated in Figs. 6.7, 6.8, 6.9, 6.10, 6.11 to determine whether the signal was performing as intended.

Fig. 6.7 shows the distortion behavior of the BD PWM signal for sinusoidal amplitude input values exceeding 0 dBFS, to verify whether the signal clips as expected when overloaded. The upper plot shows THD and THD+N in dB versus input amplitude (dBFS), while the lower plot presents the THD and THD+N in percentage versus input amplitude (dBFS). The test was performed for the BD PWM when driven by a non-quantized sinusoidal with fundamental frequency f_0=1 kHz as in-

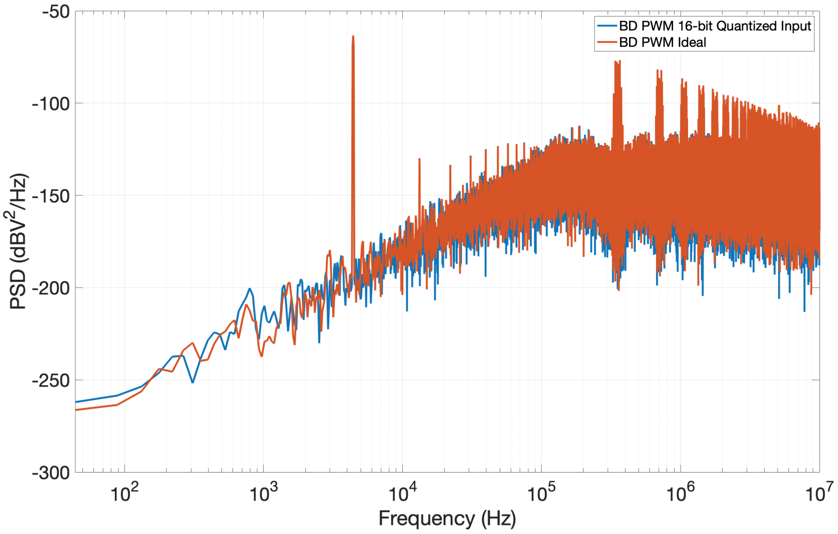


Figure 6.6: PSD of the BD PWM driven by both a non-quantized and a 16-bit quantized sinusoidal input with f_0= 4410 Hz.

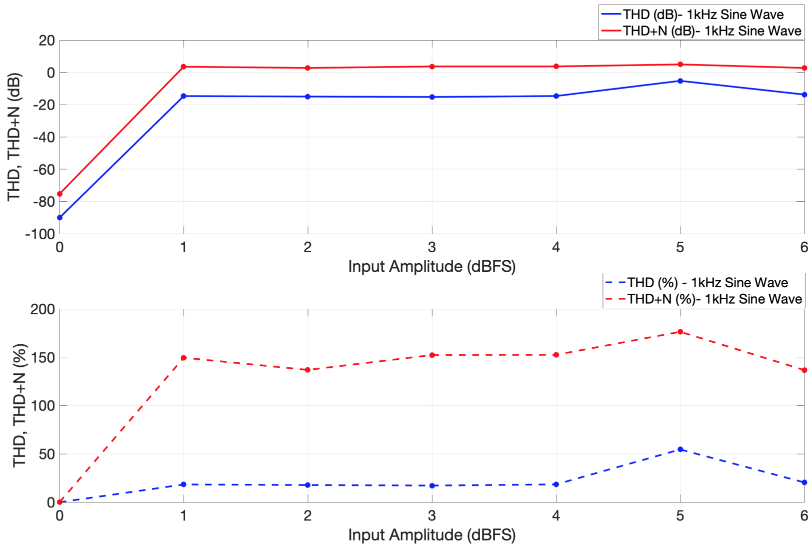


Figure 6.7: THD and THD+N of the BD PWM signal driven by a non-quantized sinusoidal input with f_0=1 kHz, for input amplitudes ranging from 0 to 6 dBFS.

put. The input amplitude of the sinusoidal was swept from 0 to 6 dBFS. Already at 0 dBFS the signal starts clipping, causing a linear increase in both THD and THD+N until 1 dBFS where it reaches an approximately constant value.

Fig. 6.8 visualizes the amount of THD and THD+N of the BD PWM when driven by a non-quantized sinusoidal with f_0=1 kHz as input, for input amplitude levels of the sinusoidal ranging from -120 to 0 dBFS. It is evident that at low input amplitude levels, THD+N exceeds 2000 %, indicating a very high level of distortion due to noise.

Fig. 6.9 shows the THD and THD+N of the BD PWM when driven by a 16-bit

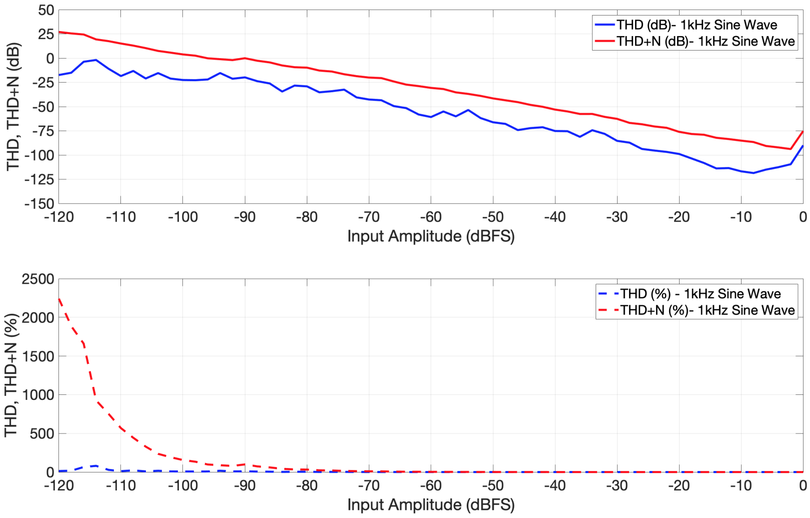


Figure 6.8: THD and THD+N of the BD PWM signal driven by a non-quantized sinusoidal input with f_0=1 kHz, for input amplitude levels ranging from -120 to 0 dBFS.

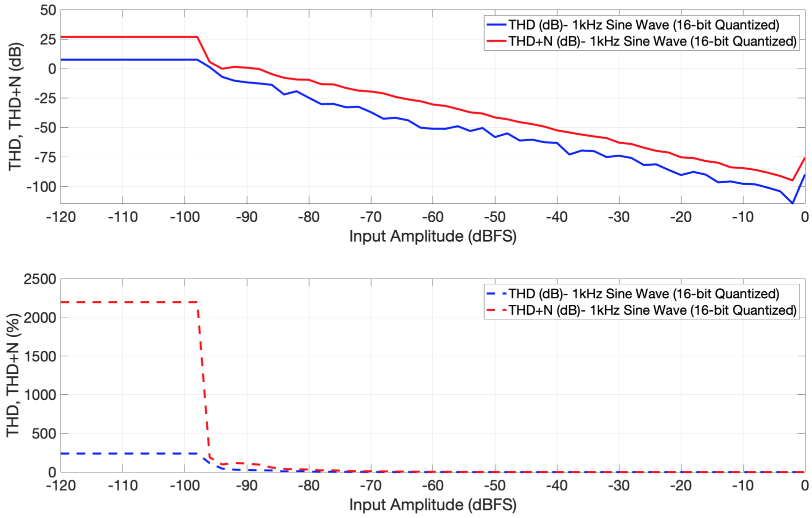


Figure 6.9: THD and THD+N of the BD PWM signal for a 16-bit quantized sinusoidal input with f_0=1 kHz, with input amplitude levels ranging from -120 to 0 dBFS.

quantized sinusoidal, with f_0=1 kHz as input, for input amplitude levels of the sinusoidal ranging from -120 to 0 dBFS. As noted in section 2.3.1, the dynamic range for a 16-bit system is around 96 dB. The plot shows that from values below -96 dBFS the THD and THD+N values are constant, indicating that the system is unable to accurately represent signals below its dynamic range.

To visualize the THD and THD+N levels of the BD PWM signal at different fundamental frequencies of the sinusoidal, a frequency sweep was performed from 20 to 9 kHz driven by a non-quantized sinusoidal input with input amplitude of -5 dBFS shown in Fig. 6.10. The upper limit of 9 kHz was chosen to ensure that the second harmonic of the signal remained within the audible frequency range (20 Hz

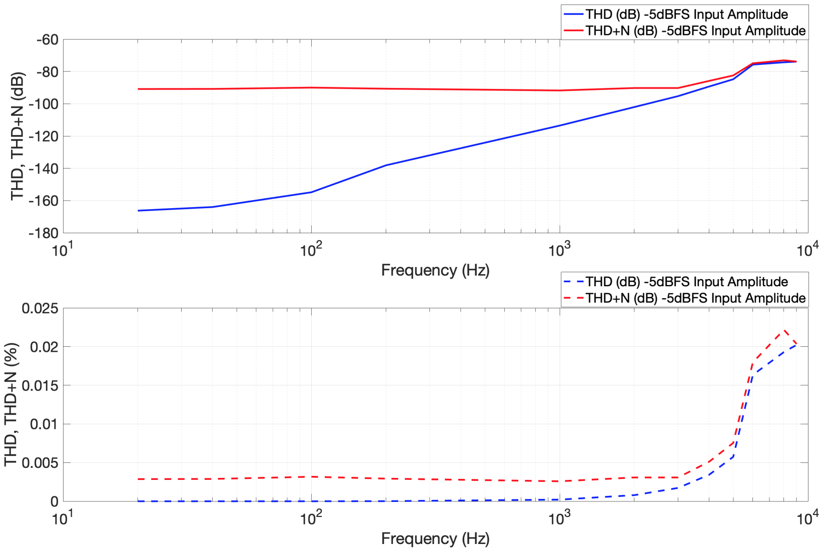


Figure 6.10: THD and THD+N of the BD PWM signal versus frequency from 20 to 9 kHz with a non-quantized sinusoidal input with -5 dBFS input amplitude.

to 20 kHz). Signals with fundamental frequencies of 10 kHz and above were omitted because the second harmonic would exceed the upper limit of the audio frequency range, making them irrelevant for this work.

The same frequency sweep was performed for the BD PWM signal when driven by a 16-bit quantized sinusoidal input with -5 dBFS input amplitude, shown in Fig. 6.11. One notable difference is that the THD for the non-quantized signal starts at approximately -165 dB at 20 Hz, while for the quantized signal it starts at around -115 dB at 20 Hz. Additionally, the THD values are slightly lower at all frequencies for the non-quantized signal.

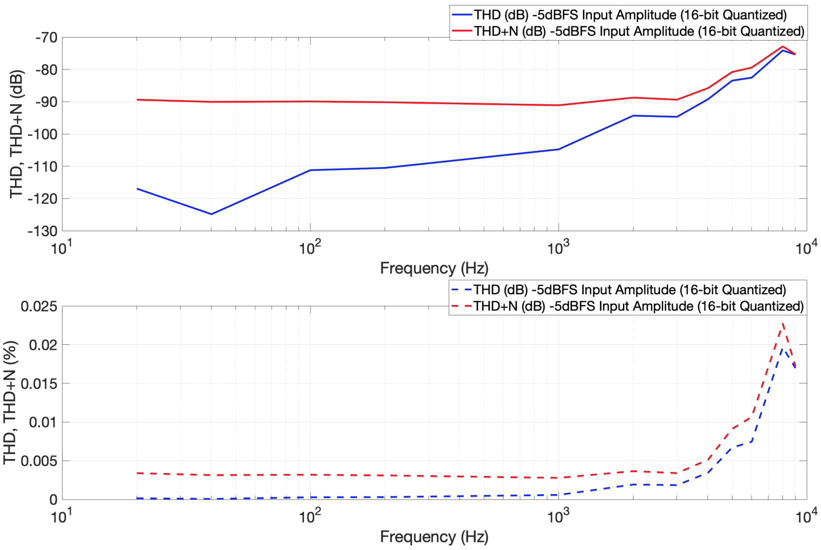
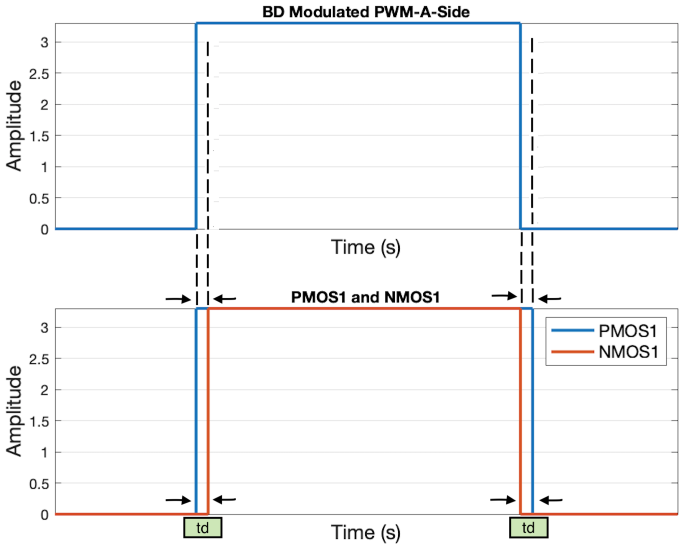


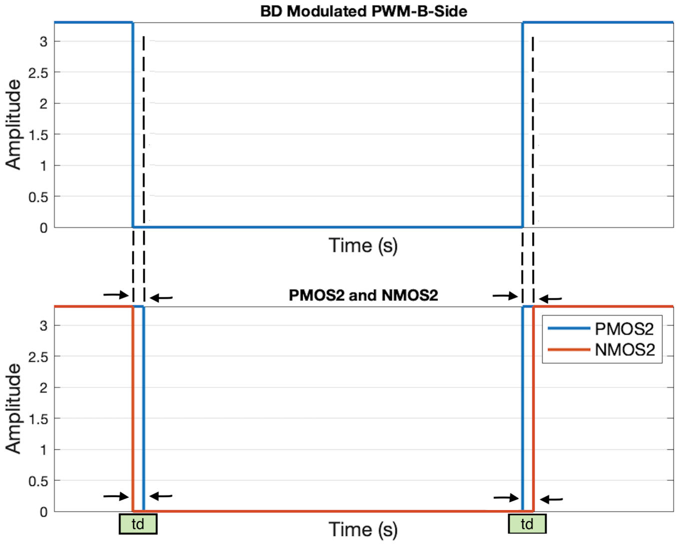
Figure 6.11: THD and THD+N of the BD PWM signal versus frequency from 20 to 9 kHz with a 16-bit quantized sinusoidal input with -5 dBFS input amplitude.

6.2.5 Implementation of Dead Time Control in BD PWM

In order to implement dead-time control in the BD PWM the principles discussed in section 3.3.1, were applied. Figs. 6.12a and 6.12b illustrate the implementation of dead time on the A- and B-sides, respectively. Since the PWM clock frequency was set to $f_{\mathrm{PWM}} = 90.3168$ MHz, the corresponding minimum dead time is approximately $t_{\mathrm{dead,min}} = 1/f_{\mathrm{PWM}} \approx 11$ ns.



(a) Zoomed view of added dead time on A-Side with PMOS1 and NMOS1



(b) Zoomed view of added dead time on B-Side with PMOS2 and NMOS2

Figure 6.12: Implementation of Dead time on A- and B-Sides. Zoomed-in views illustrating the added dead time between the high-side (PMOS1/PMOS2) and low-side (NMOS1/NMOS2).

6.2.6 Dither Noise Generation in Quiescent Power Measurement

The quiescent power consumption of the power amplifier, P_Q, is in this work defined as the power loss when the amplifier is in a quiescent state and calculated using (2.15). This represents a scenario when there is no audio input signal present, only background noise with an amplitude bounded within ± 1 LSB or $\pm\, 1/(2^{15} - 1)$.

To measure the quiescent power consumption of the power amplifier, the theory regarding single sided BD modulation was employed as introduced in section 3.2.1. Instead of using two sinusoidal input signals, two dithering triangular noise signals were used, visualized in Fig. 6.13. They were generated by summing two independently uniform random signals.

Fig. 6.14 shows the power spectral density of the BD PWM signal when driven with triangular noise. The spectrum primarily consists of the $\Delta\Sigma$ shaped noise floor and the PWM switching harmonics, at the switching frequency. No fundamental harmonic components can be seen, as the input only consists of triangular noise.

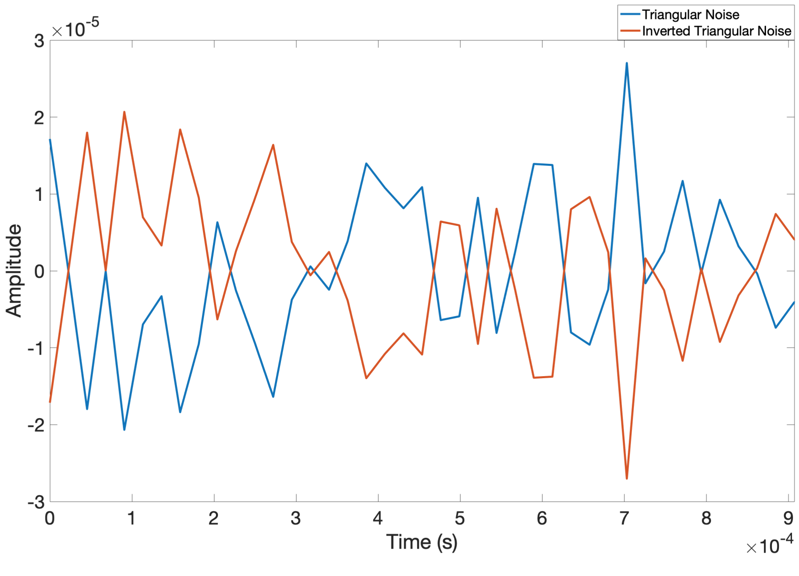


Figure 6.13: Time-domain representation of the triangular noise and the inverted triangular noise.

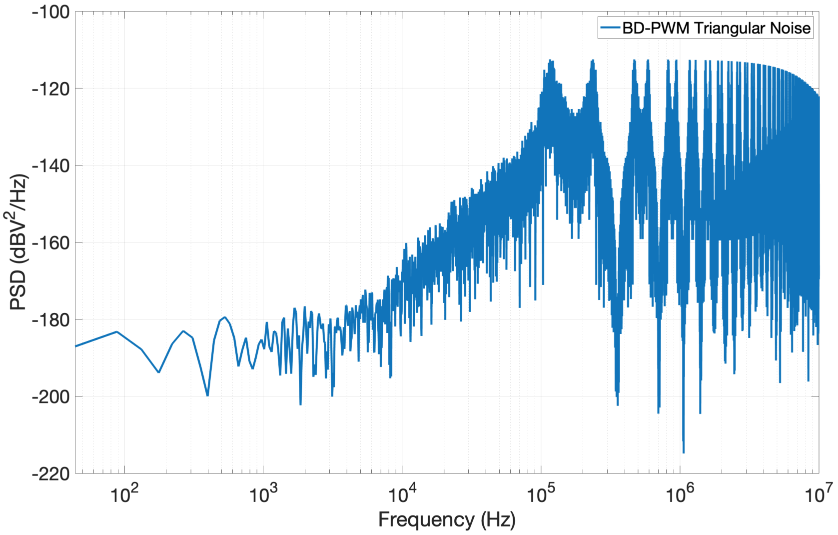


Figure 6.14: PSD of the BD PWM signal driven by triangular noise.

6.3 Design and Simulation of the Class D Power Amplifier in LTspice

For the circuit simulations of the Class D power amplifier, LTspice was chosen due to its simplicity, efficiency and relatively accurate modeling of circuit components. The MOSFET gate control signals were first generated in Matlab, resampled and then imported into LTspice as voltage sources. The selected MOSFETs for the PCB, namely PMDXB950UPEL and PMDXB290UNE could not be imported into LTSpice [42, 43]. Thus, a suitable MOSFET pair with similar switching parameters parameters was found, as shown in Table 6.2. For the PMOS pair, Si1555DL$_\mathrm{P}$ was selected, and Si1555DL$_\mathrm{N}$ was selected for the NMOS pair [44].

Fig. 6.15 visualizes the output stage of the Class D power amplifier implemented in LTspice using an H-bridge circuit, with the corresponding component values listed in table 6.3. It was designed without an output filter, using a series RLC circuit as discussed in section 3.3.2, which consisted of an inductor L_1 or L_2 in series with its internal serial resistance R_L, connected in series with the piezoelectric actuator C_Piezo. Since the piezoelectric actuator has both capacitive and resistive characteristics it was modeled using both a capacitor and a resistance. In technical literature regarding Class D power amplifiers with piezoelectric actuators, the piezoelectric

Table 6.2: Comparison of the switching parameters of the MOSFETs on the PCB and in LTspice.

MOSFET	$\mathbf{Q_g}$ (Charge)	$\mathbf{R}_{\mathrm{DS(on)}}$
Si1555DL_N	8×10^{-10} C	0.58 Ω
Si1555DL_P	1.5×10^{-9} C	0.63 Ω
PMDXB950UPEL	0.9×10^{-10} C	0.32 Ω
PMDXB290UNE	1.8×10^{-9} C	1.00 Ω

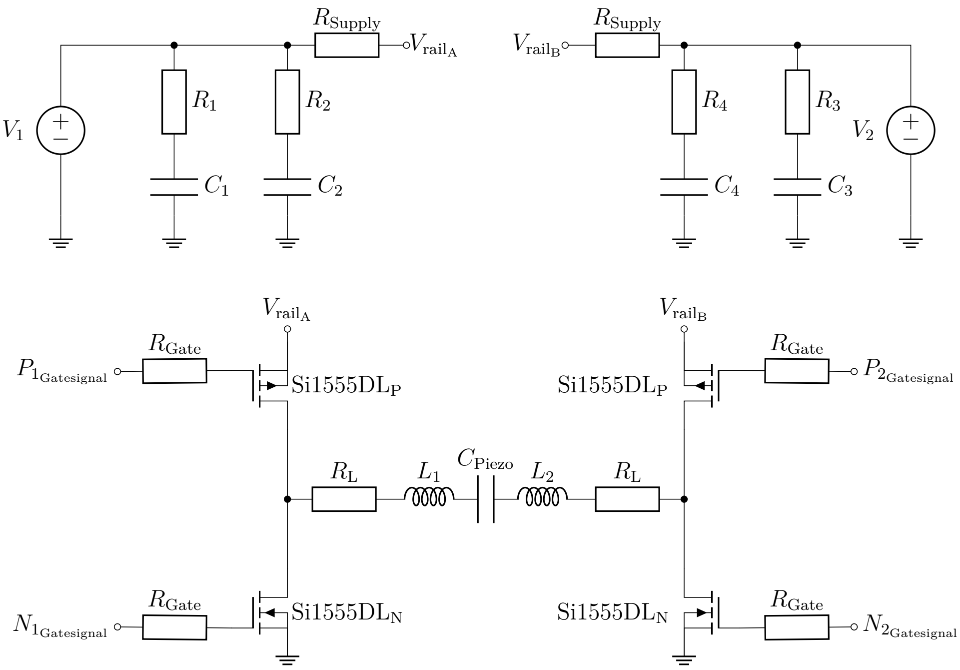


Figure 6.15: Output power stage of the Class D power amplifier implemented as an H-bridge circuit and simulated in LTspice.

actuator typically has a capacitance value between 1 and 4 μF as listed in Table 8.1. Thus, the capacitance C_{Piezo} was selected as 1 μF, with a serial resistance of 10 mΩ, and the chosen inductance of the inductors L_1 and L_2 were selected as 47 μH with a serial resistance of 0.47 Ω. This configuration of component values in the series RLC circuit, resulted in a resonance frequency of 23.215 kHz as calculated by (3.4), and a Q factor of approximately 12.03 determined by (3.5). The bode plot depicting the magnitude and phase response over the piezoelectric actuator is shown in Fig. 6.16.

To simplify the simulations the output stage was designed using two separate voltage sources V_{rail_A} and V_{rail_B} as shown in Fig. 6.15, instead of a single supply as on the PCB. Each voltage source included two series low-pass RC filters and a series resistor

Table 6.3: List of circuit components and their values.

Component label	**Value**
R1, R3	0.3 Ω
R2, R4	1 mΩ
L1, L2	47 μH
C_{Piezo}	1 μF
R_L	0.245 Ω
C1, C3	100 μF
C2, C4	10 μF
V1, V2	3.3 V
R_{Supply}	10 mΩ
R_{Gate}	10 mΩ

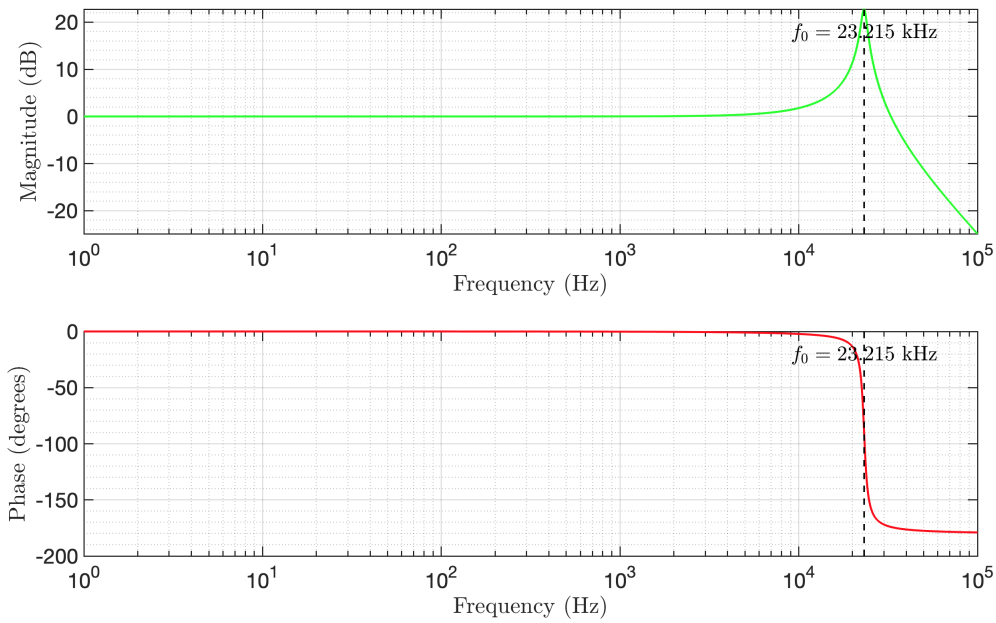


Figure 6.16: Frequency response of the RLC filter.

R_{Supply}. The first filter had a cut-off frequency of around 5.3 kHz, and consisted of R_1 and C_1 on V_{rail_A} and R_4 and $C4$ on V_{rail_B}. The second filter with a cut-off of around 16 MHz, consisted of R_2 and C_2 for V_{rail_A} and R_3 and C_3 for V_{rail_B}.

6.4 Design of the Output Stage PCB

For the development and design of the output stage PCB, KiCad was used due to its simplicity to use and the arrangement of features. Fig. 6.17 presents the schematic of the PCB, where the selected circuit components and their values are listed in Table 6.4. Ceramic multilayered decoupling capacitors, $C_1 - C_6$ were placed on the PCB to minimize parasitic inductance and noise. In series with each microcontroller output pin, a 0 Ω ballast resistor $(R_1 - R_7)$ with a generic footprint was placed, allowing for switching to a different resistance value if necessary. On the 2-layered PCB, the top layer was connected to V_{DD} and the bottom layer was connected to ground GND.

Table 6.4: List of selected circuit components and their values.

Component	Label	Value
GRT188R61H105KE13D	C_1, C_4	1 μF
GRM188R6YA106MA73J	C_2, C_5	10 μF
GRM188R60J476ME01D	C_3, C_6	47 μF
PMDXB950UPEL	Q_1	-
PMDXB290UNE	Q_4	-
LPS6235-223MRC	L_1, L_2	47 μH
Resistor	$R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8$	0 Ω

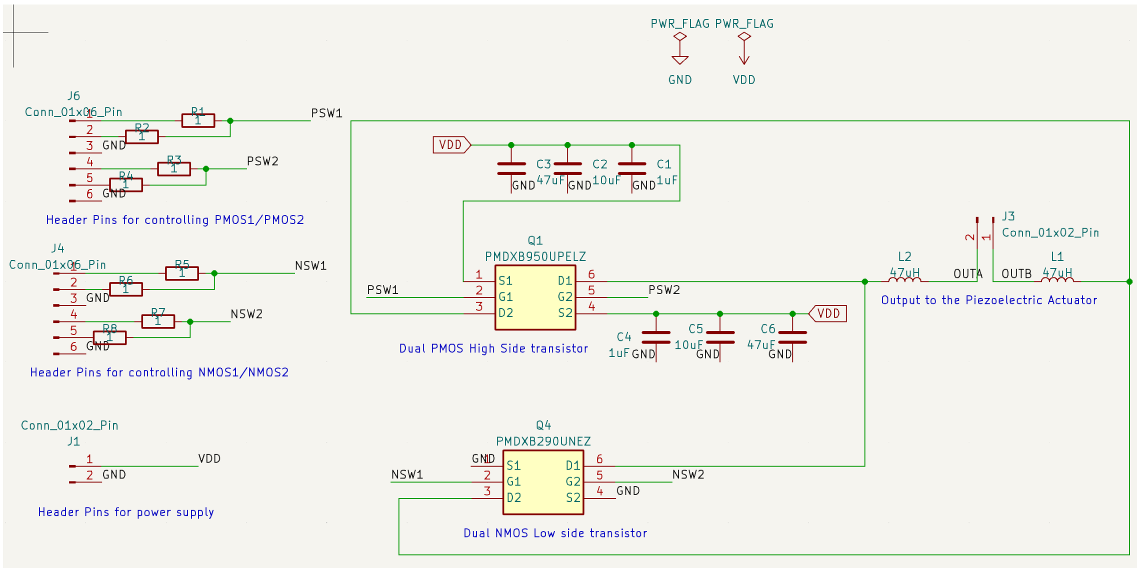


Figure 6.17: Schematic of the PCB layout designed in KiCad.

6.5 Microcontroller Implementation

Next step was to implement the DSP system tested in MATLAB on the microcontroller. Some of these steps were carried out in parallel with the MATLAB coding. The coding on the LPC55S69 was carried out on its designated devkit LPCXpresso55S69-EVK.

6.5.1 Coding

The approach to implement the second order $\Delta\Sigma$ modulator on the microcontroller was similar, but there were some key differences. For example, the microcontroller uses the C language, and the specific data type have a great impact when it comes to timing and execution of the program. The first step was to translate the code from MATLAB to C. The second order $\Delta\Sigma$ modulator structure, along with the interpolation stage was implemented, and the FIR filter coefficients were imported along side a sinusoidal test signal. The test signal was a sinusoidal tone at 400 Hz, rescaled to an 8-bit integer.

6.5.2 I^{2}s

There are two ways of implementing I^{2}s on the LPCXpresso55S69-EVK. Either using dedicated pins or by using the audio stream line located at the top of the controller. First, an attempt was made to use the pins to receive audio, by generating an audio wave using an audio analyzer. However, this was unsuccessful and some errors in the microcontroller could not be resolved. Therefore, a demo using the audio jack was adopted. Using this code, audio was received and handled using a ping-pong type buffer. However, due to time constraints, a sinusoidal tone was imported in order to start with PWM generation.

6.5.3 PWM

The microcontroller has several ports that can be used to output PWM. For this project, four pins are needed, one for each MOSFET, and eight are preferred in order to generate a two times higher current to drive the gates without using a gate driver circuit. Luckily, the LPCXpresso55S69-EVK has 9 different PWM output ports, and all can be controlled using a single timer, SCT0.

To find which pins can be used in the devkit for PWM you first have to look up the SCTimer output pins of the sctimer, in Table 501 in [45]. Following this, you have to go into the schematic diagram at sheet 3 of [46] and see where the pins are located in the schematic diagram. After this, look at the devkit user guide [47], where you will be able to locate the outputs.

7

Results

This chapter presents the results of the thesis. It begins with a performance evaluation of the SNR and SINAD of the second-order $\Delta\Sigma$ modulator. Next, the $\Delta\Sigma$ to BD PWM conversion is analyzed, followed by an analysis of the dead-time control implementation and its effect on the output signal in the form of shoot-through current elimination and distortion. After that, a 3D design of the printed PCB prototype is presented. Finally, the Class D power amplifier's performance is assessed based on key parameters including THD, THD+N, SNR, SINAD, power consumption, and apparent power efficiency.

7.1 Performance of the $\Delta\Sigma$ modulator

The SNR and the SINAD for the second-order 7-bit $\Delta\Sigma$ modulator was analyzed for input signals with different input signal levels, both with and without 16-bit quantization, using (2.1) and (2.5). Fig. 7.1 visualizes the SNR and SINAD vs different input signal levels.

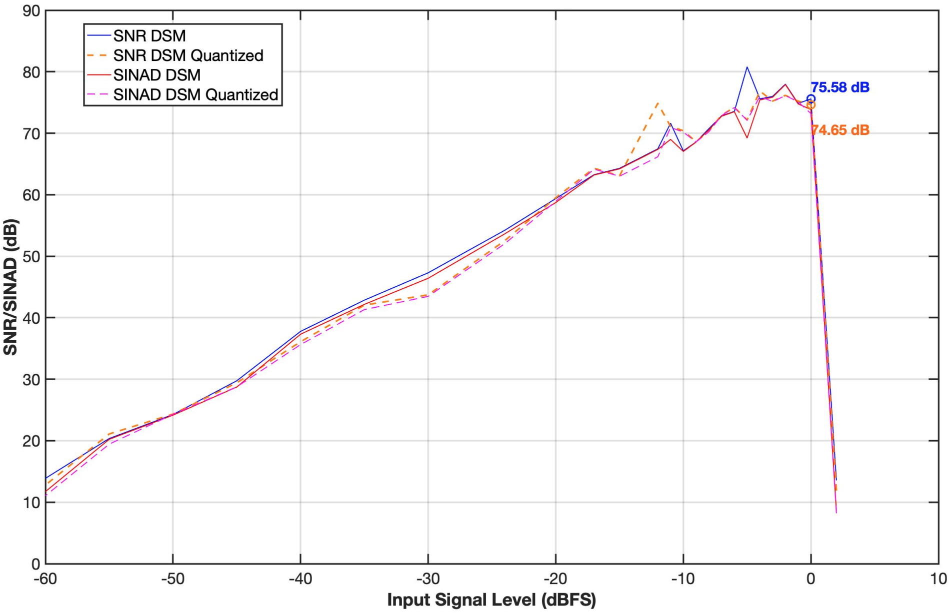


Figure 7.1: SNR and SINAD versus input signal level for a second-order 7-bit $\Delta\Sigma$ modulator, driven by a sinusoidal with and without 16-bit quantization for different input signal levels at $f_0 = 4410$ Hz.

The theoretical maximum SNR for a second-order ΔΣ modulator is given by (5.9). For a 0 dBFS input, the maximum SNR is 76.1597 dB. As shown in Fig 7.1, the maximum measured SNR for the second order ΔΣ modulator is 75.58 dB with a non-quantized input and 74.65 dB with a 16-bit quantized input, both very close to the theoretical max. The SINAD values are very close to the SNR values and approximately the same for both with non-quantized and with 16-bit quantized input.

7.2 Analysis of ΔΣ to BD PWM Conversion

Before implementing the dead-time control into the BD PWM signal, the signal performance of the BD PWM without dead time had to be evaluated. In Fig. 7.2, the relationship between SNR/SINAD and the BD PWM signal of different input signal levels is illustrated. As shown, for a 0 dBFS input, the SNR of the BD PWM is 79.37 dB with a non-quantized input and 78.97 dB with a 16-bit quantized input. Due to the additional distortion in the ΔΣ to BD PWM conversion, the SINAD is slightly lower than the SNR for both the BD PWM with non-quantized and with 16-bit quantized input.

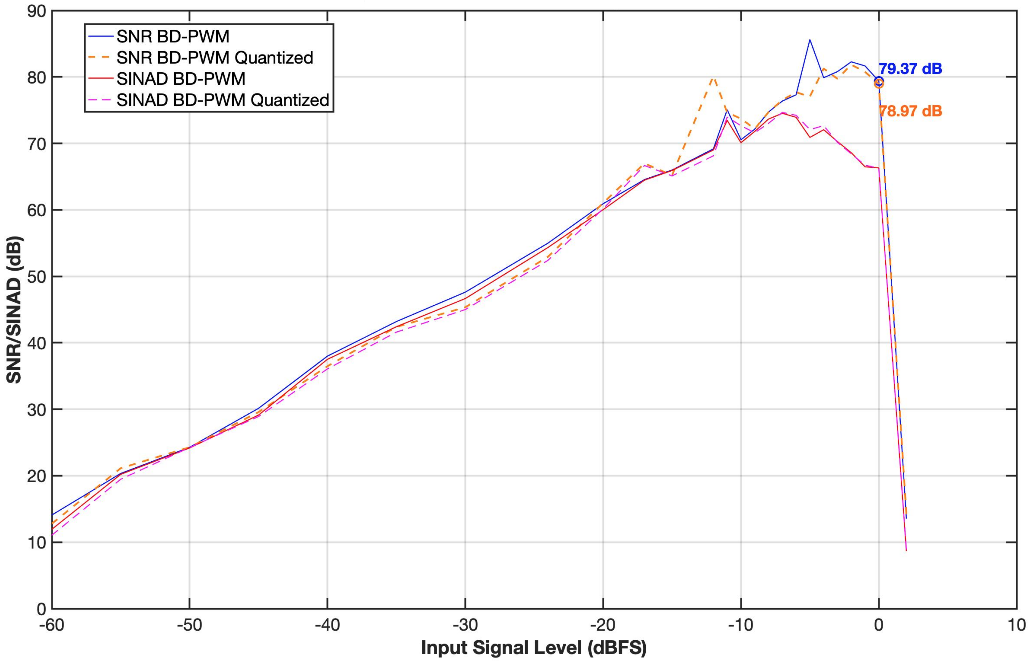


Figure 7.2: SNR and SINAD versus input signal level for the BD PWM output, driven by a sinusoidal input with and without 16 bit quantization at f_0= 4410 Hz.

7.3 Dead-Time Control Optimization and Shoot-Through Prevention

Simulations of the dead-time control were conducted on the H-bridge circuit shown in Fig. 6.15 with a BD PWM with varying dead times, driven by a sinusoidal input at 0 dBFS with $f_0 = 4410$ Hz, to visualize both the shoot-through current caused by insufficient dead time and the distortion caused by excessive dead time. The result is illustrated in Fig. 7.3.

Fig. 7.3 presents two plots, containing three signals each. The upper plot 7.3(a) depicts the output voltage of the BD PWM, over one period. A time period proportional to the added dead time is inserted at the beginning and end of each cycle. This generates distortion, as the PWM signal no longer fully represents the intended output signal. Alongside this issue, dead time also causes a voltage drop proportionate to the diode voltage of the transistor. The bottom plot Fig. 7.3(b) shows the switching currents of the MOSFETs. Without dead time, the switching current is significantly higher due to the shoot-through current. When increasing the dead time, shoot-through is eliminated. The distortion is also evident in this plot, as the switching current is shifted to the right in time. This plot confirms that 11 ns of dead time is enough to avoid shoot-through.

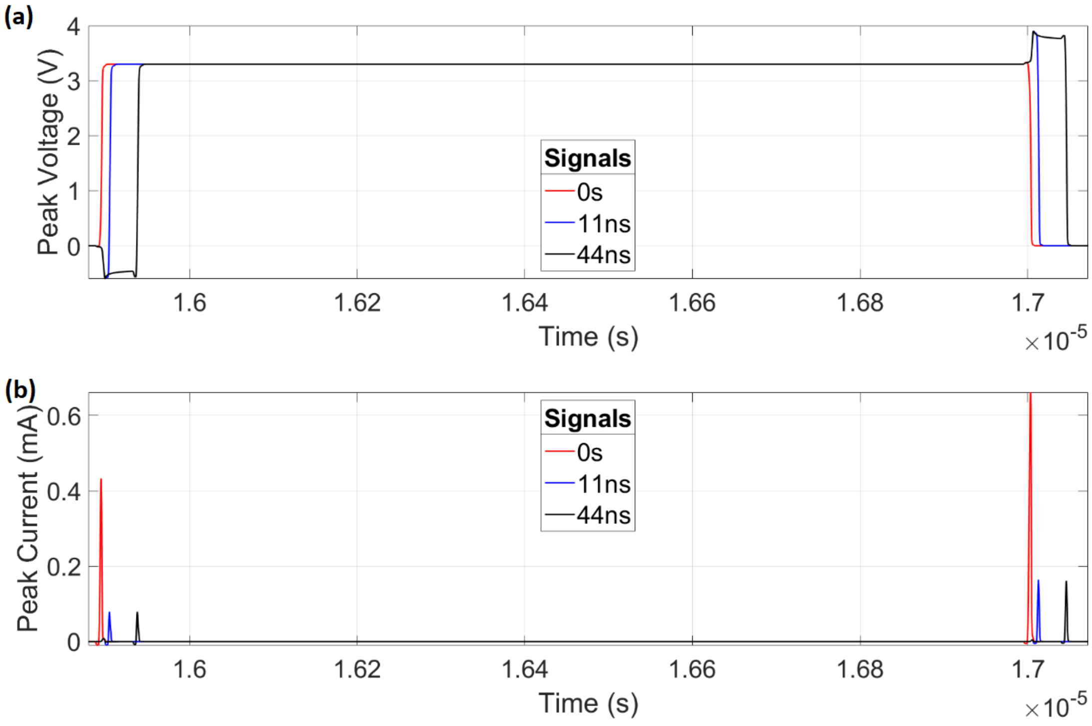


Figure 7.3: **(a)**BD PWM output voltage using different dead times. **(b)**Switching current at different dead times.

Fig. 7.4, shows the relationship between dead time (x-axis) and SINAD (y-axis) for a BD PWM driven by a sinusoidal input signal at 0 dBFS with f_0=4410 Hz. The optimal SINAD is obtained with absolutely no dead time, but that would cause a shoot-through current that might harm system components, as seen in Fig. 7.3. According to the circuit's specifications and previous tests, it can be concluded that 11 ns of dead time is enough to reduce the shoot-through current while keeping distortion to a minimum. Using a dead-time longer than 11 ns results in an almost linear decrease in SINAD.

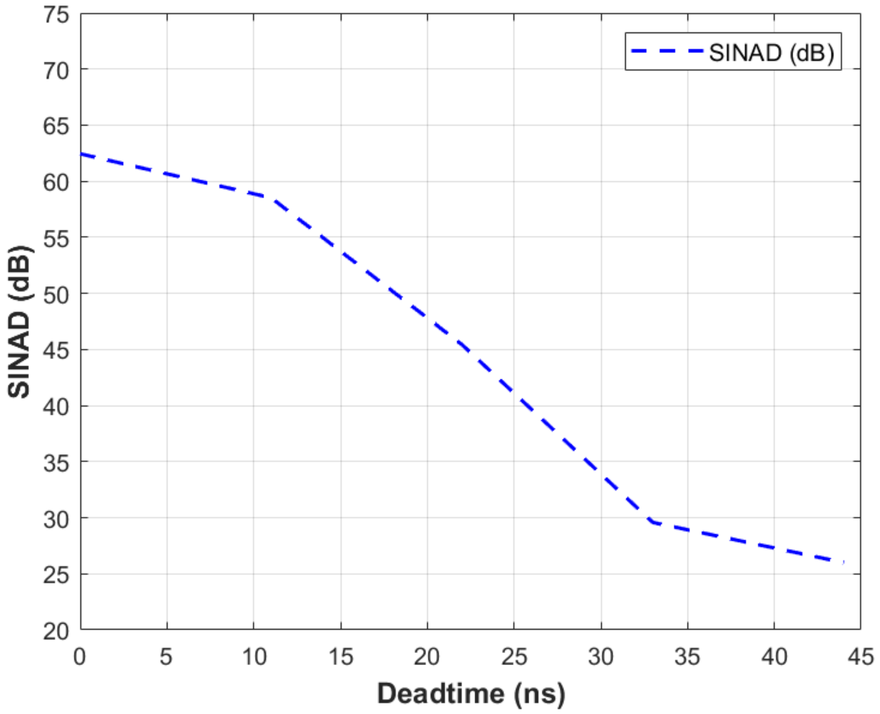


Figure 7.4: SINAD versus dead time for a BD PWM driven by a 0 dBFS 16-bit quantized sinusoidal input at f_0=4410 Hz.

7.4 PCB Design and Prototype

The final design of the printed PCB is shown in Fig. 7.5. It is 3D-rendered using the circuit displayed in Fig. 6.17. When connecting the LPCXpresso55S69-EVK microcontroller devkit to the PCB, the header pins J4 on the left side are used for one half of the microcontroller's output pins, while the remaining output pins are connected to header pins J6 on the right side.
The header pins J1 are used for the connection of ground and supply voltage V_{DD}, while the output signal from the H-bridge is taken from header pin J3 situated in the middle of the PCB. This is where the piezoelement should be connected.

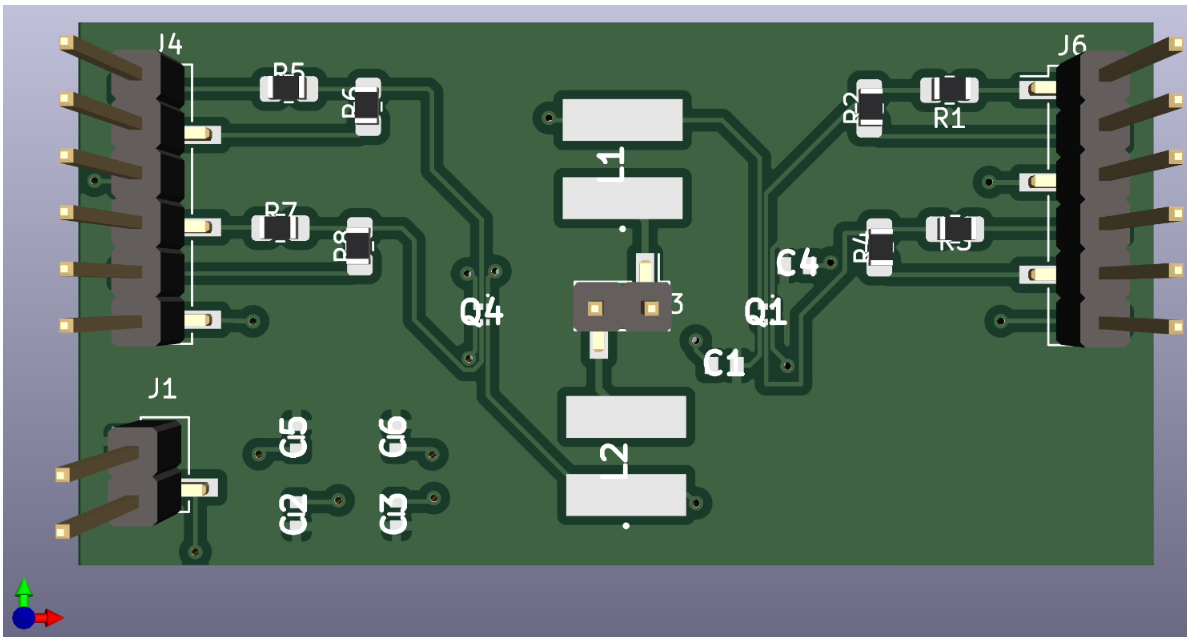
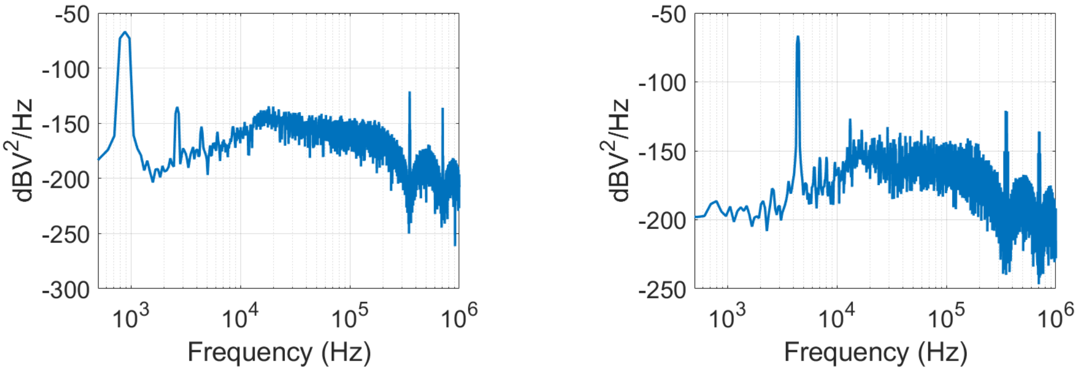


Figure 7.5: A 3D render of the PCB in KiCad.

7.5 Frequency Domain Analysis: SNR, THD, and THD+N

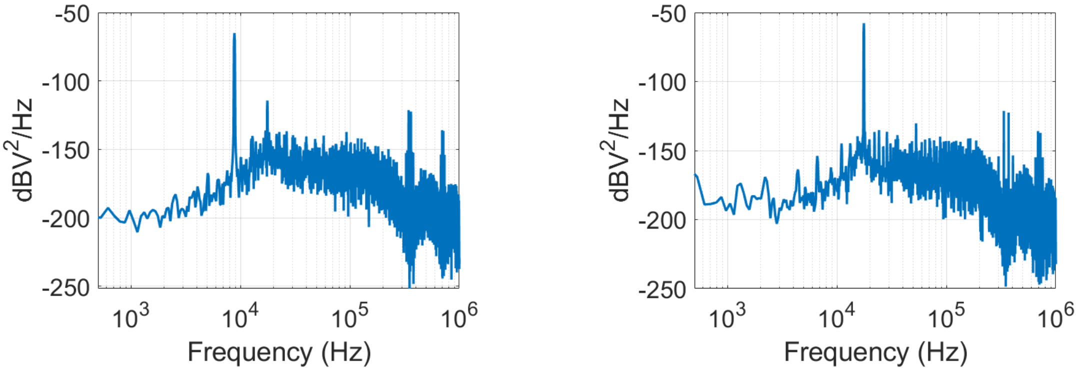
Figs. 7.6a, b, c, and d illustrate the PSD of the power amplifier output for sinusoidal inputs at four different fundamental frequencies within the audible range. The selected fundamental frequencies were 882 Hz in Fig. 7.6a, 4410 Hz in Fig. 7.6b, 8820 Hz in Fig. 7.6c and finally 17640 Hz in Fig. 7.6d.

The fundamental frequency component can be distinguished as the main tone in each of the PSD spectra. However, at lower frequencies, the tone appears wider. As an example, note the main tone at 882 Hz in Fig 7.6a and its corresponding first harmonic in comparison to the other spectra in Fig 7.6. The difference between the



(a) PSD of the power amplifier output using a sinusoidal tone with $f_0 = 882\,\text{Hz}$ at -11 dBFS and 11 ns dead time.

(b) PSD of the power amplifier output using a sinusoidal tone with $f_0 = 4410\,\text{Hz}$ at -11 dBFS and 11 ns dead time.



(c) PSD of the power amplifier output using a sinusoidal tone with $f_0 = 8820\,\text{Hz}$ at -11 dBFS and 11 ns dead time.

(d) PSD of the power amplifier output using a sinusoidal tone with $f_0 = 17640\,\text{Hz}$ at -11 dBFS and 11 ns dead time.

Figure 7.6: PSD of the power amplifier output at fundamental frequencies 882, 4410, 8820, and 17640 Hz with a sinusoidal input signal of -11 dBFS and an 11 ns dead time.

spectra arises from the scaling of the x-axis. The frequency resolution depends on the total simulation time in LTspice, and the tone would appear more slim if the simulation time were increased. The RLC filter resonance peak is at 23.215 kHz, and causes amplification in the frequency range of 16-20 kHz, which is visible on each plot. Following this, there is a decline in amplitude in the spectrum, corresponding to the stop band of the filter. At the end of each plot, the large peaks are the switching harmonics at the frequency of the BD-PWM.

Fig. 7.7 shows a THD+N frequency sweep for the output signal of the simulated power amplifier circuit, with a dead time of 11 ns and an input signal level of -11 dBFS. The frequency sweep was performed using sinusoidal signals with fundamental frequencies in the hearing spectrum (20-20 kHz).

The bandwidth of our signal is the same as the hearing spectrum (20 Hz to 20 kHz), meaning that frequency components below 20 Hz and above 20 kHz are considered irrelevant. The THD+N is below 0,2% until it around 5 kHz, where it starts to increase drastically until around 9 kHz where it peaks before greatly decreasing.

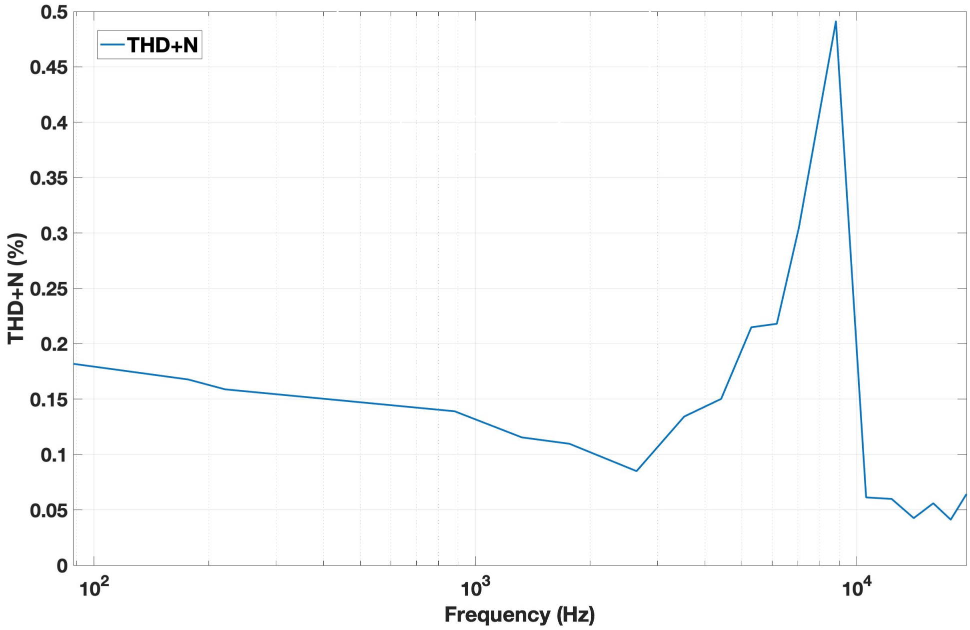


Figure 7.7: THD+N frequency sweep of the simulated power amplifier, driven by a sinusoidal with input signal level of -11 dBFS and 11 ns dead time.

Fig. 7.8 shows an input signal level sweep of the simulated power amplifier, showing the resulting SNR/SINAD to visualize how it handles different input signal levels and observe the distortion when being overloaded.

The SNR and SINAD values greatly increase with the input signal level, rising from their minimum values of 2 dB for SNR and 5 dB for SINAD at -60 dBFS to a peak SNR of 69.3 dB at -12 dBFS and a peak SINAD of 58 dB at -11 dBFS. As shown in the figure, the power amplifier operates under optimal conditions when driven by an input signal level between -16 and -5 dBFS.

Lastly, the THD and THD+N are compared with the RMS output voltage V_{RMS} in Fig. 7.9. The lowest THD+N value was found to be 0.15% at 0.36 V_{RMS}, and the lowest THD value was approximately 0.055% at 0.36 V_{RMS}.

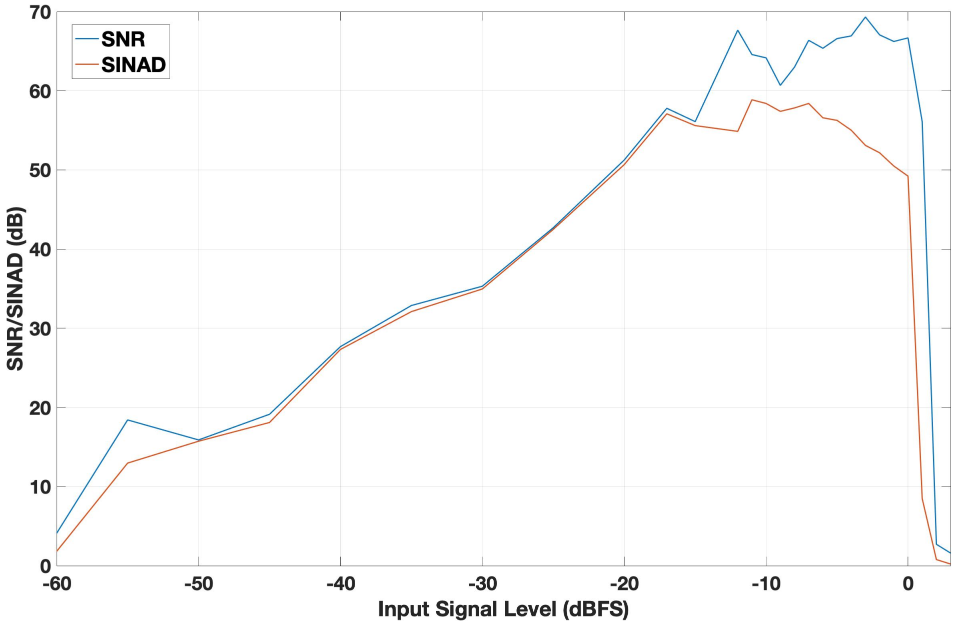


Figure 7.8: Input signal level sweep showing the resulting SNR/SINAD of the simulated power amplifier.

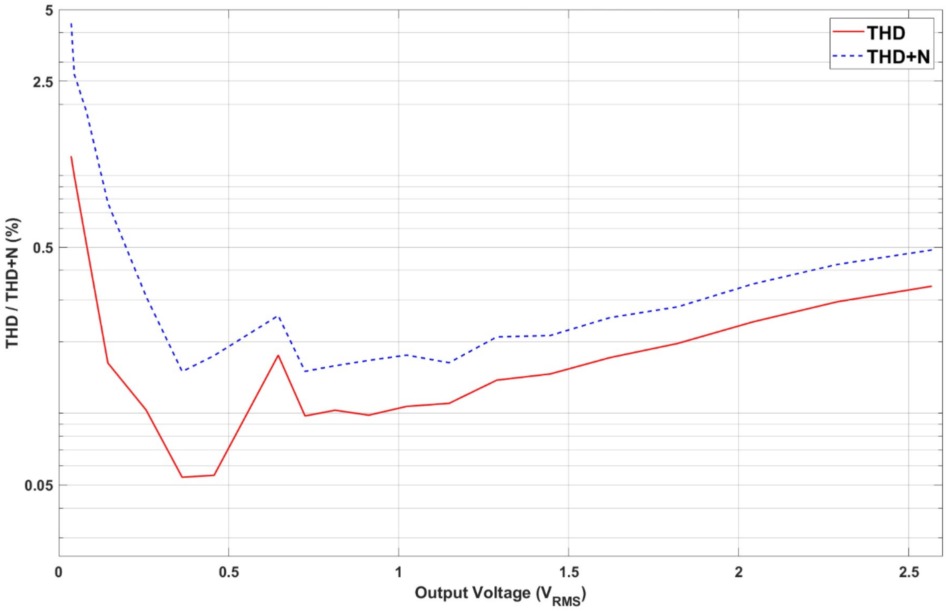


Figure 7.9: THD/THD+N versus RMS output voltage V_{RMS}.

7.6 Power Consumption and Efficiency

An overview of the resistive power losses in the simulated power amplifier is shown in Fig. 7.10. The blue curve in the figure illustrates the power losses over the inductor calculated as the product of the RMS current over the inductor I_{o} and its serial resistance R_{inductor} calculated as: $I_{\mathrm{o}}^2 \times R_{\mathrm{inductor}}$. The red curve shows the power losses of the on state resistance, as given by (2.16), and the purple curve shows the power losses of the total resistance R_{tot}, where the total resistance is the sum of R_{inductor} and $R_{\mathrm{DS(on)}}$, calculated by: $I_{\mathrm{o}}^2 \times (R_{\mathrm{inductor}} + R_{\mathrm{DS(on)}})$. The green curve shows the quiescent power losses as given by (2.15).

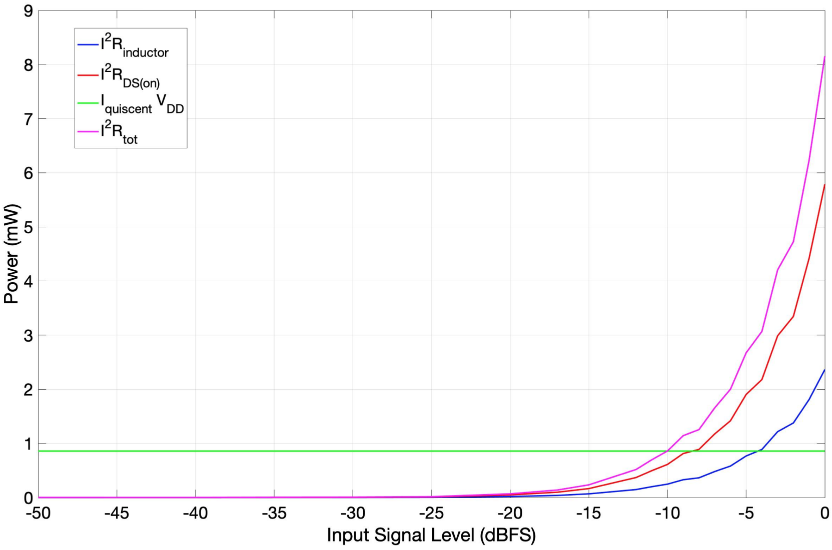


Figure 7.10: Overview of the resistive and quiescent power losses in the audio power amplifier.

As shown in the figure, the power losses increase significantly at higher input signal levels, which corresponds to a higher output voltage.

Fig. 7.11 shows the apparent power efficiency, as given by (2.13), versus different output RMS voltage levels. The apparent power efficiency increases with the output RMS voltage. The highest achievable efficiency was 86.4%. At 0 dBFS, the output RMS voltage is around 2.33 V, while it is merely a few mV at -60 dBFS, hardly enough to drive the amplifier and the load efficiently.

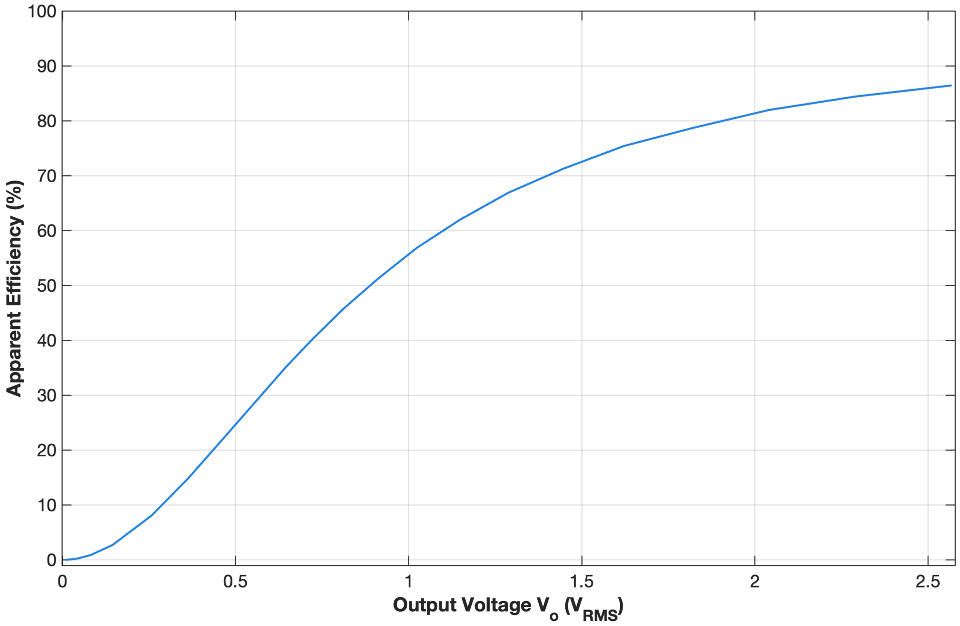


Figure 7.11: Apparent power efficiency versus RMS output voltage for the simulated power amplifier.

8

Discussion

In this chapter, the results of the thesis will be discussed. The results of the different measurements and their outcomes will be analyzed, and how the results could be improved upon. To provide context, Table 8.1 compares the results of this thesis with technical literature as well as commercial products.

Table 8.1: Performance Comparison of Technical Literature and Commercial Class D Audio Amplifiers for Bone Conduction Hearing Aids with Piezoelectric Loads.

Parameter	This Work	[16]	[48]	[49]	[50]	[51]	[52]
V_{out} (V_{pp})	7.1	18	14.4	14.4	19	24	20
THD+N$_{\text{PEAK}}$ (%)[a]	0.15	0.025	0.0043	0.0039	0.070	0.0132	0.100
THD+N$_{\text{PEAK}}$ (dB)[a]	−56.48	−72.04	−87.3	−87.2	−63.10	−77.6	−60
Piezo Load (μF)	1	0.47	4	4	1	4	–
I_Q (mA)	0.24	0.7	8.5	8.7	4	10.9	13
P_Q (mW)	0.795	6.3	122.4	–	22	–	48
SNR (dB)	69.32	95	105.8	–	94	111	80
F_s (kHz)	352.8	750	–	–	300	768	250
$\hat{P}_{App}$ (%)	86.4	96	–	–	92	–	90

[a] Measured with an input of 1 kHz; for this work the input was 1.323 kHz.

The maximum SNR in our work was measured to be 69.32 dB, which is significantly lower than some of the products mentioned in Table 8.1. The SNR after the $\Delta\Sigma$ modulator is limited according to (5.9), and depends on the oversampling rate, order of the modulator, and the output word length. The highest SNR measured after the modulation stage was 74.65 dB, which is close to the theoretical maximum of 75.1597 dB. Therefore, one of the aforementioned parameters has to be changed to increase the SNR after the modulation stage.

In our case, the OSR was limited to 8, resulting in a sampling frequency of $F_s = 8\times 44.1\text{ kHz} = 352.8\text{ kHz}$, where 44.1 kHz is the standard sampling rate for audio applications. Although a higher OSR would further reduce in-band noise, it was important to align the frequencies of the $\Delta\Sigma$ modulator and PWM switching speed below 500 kHz. Having the PWM frequency below 500 kHz is beneficial, as it maximizes the PWM-cycle, i.e., period, while still maintaining a reasonably high switching frequency.

Adding a dead time to each switching instance causes signal distortion. Signal information is lost at both edges, as the signal is forced to switch polarity earlier or later than intended. For example, when using 11 ns of deadtime, 11 ns of information is lost at every switching instance. When using a sampling rate F_s of 352.8 kHz, the PWM-cycle is calculated as $1/F_\mathrm{s}$, resulting in approximately $1/(352.8 \times 10^3) \approx 2.83\ \mu$s. In contrast, with an OSR of 128, resulting in a F_s of 5.6448 MHz, the PWM-cycle would be approximately $1/(5.6448 \times 10^6) \approx 177$ ns. Adding a dead time of 11 ns to both the rising and falling edges (as shown in Fig. 6.12) causes far less distortion for a signal with a 2.83 μs cycle than a signal with 177 ns. Thus, while a higher OSR could improve the SNR in the modulation stage, it could also cause more distortion in the output stage, degrading the SINAD and reducing sound quality.

Considering the OSR's effect on the output stage, the only way to increase our SNR would be to implement a higher-order $\Delta\Sigma$ modulator, or use a longer word length. These changes could increase the SNR after the modulation stage, with the hope of also improving output-stage performance. A higher modulator would perform better, but it would also make the modulator unstable, making balancing tricky. Therefore, if time is not a constraint, focus should be put on designing a higher order modulator.

Another possible approach would be to increase the output word-length from the modulator. However, one of the purposes of a $\Delta\Sigma$ modulator is to reduce the word length to reduce computational strain on the microprocessor, and hardware complexity if the modulator is implemented as hardware. Therefore, the implementation of this change would rely on the quality of the processor, as well as the speed requirements. If computational power and speed are not an issue, then it would be beneficial to increase the output word length in the $\Delta\Sigma$ modulation stage. But as with all these changes, even if the SNR is increased in the modulation stage, it does not guarantee a higher SINAD or SNR after the output stage.

The dead time could also be reduced to increase the SNR. Taking a look at Fig. 7.3(a), the higher the dead time, the higher the distortion, reducing audio quality and SINAD. This is also illustrated in Fig. 7.4, concluding that dead time should be as low as possible. However, hardware specifications limit the dead time to 11 ns, and using a shorter dead time could risk causing shoot-through if a MOSFET would malfunction and conduct for an abnormally long period, illustrated in the bottom figure of Fig. 7.3.

The lowest measured THD+N was -56.48 dB. Comparing our measured THD+N with other works in Table 8.1, it is significantly higher. The THD+N is closely related to the SNR and SINAD. Thus, it is limited to the SNR of the modulated input signal, in combination with the distortion caused by the hardware itself. The THD+N could decrease by imposing the changes as suggested in the paragraph above, but without further testing, it is impossible to assess the hardware circuit's effect on the THD+N. Furthermore, it is difficult to assess the damage the THD+N and THD have on the system without being able to listen to the audio. If the distortion is distinguishable, it poses a great problem to sound quality.

8.1 Frequency analysis

Fig. 7.6 displays the output power spectrum using 4 different fundamental sinusoidal tones: 882 Hz, 4410 Hz, 8820 Hz, and 17640 Hz. These are all either multiples, or factors of 4410, which is a factor of the original sampling frequency of 44.1 kHz. These frequencies were used to minimize spectral leakage. The graphs compare the PSD of the output signal, which shows the power of the different spectral components in the output signal, such as the fundamental component.

The tones are all individually distinguishable at the correct frequencies, along with the distortion caused by the input signal, visible at multiples of the fundamental frequency. The switching distortion caused by converting the output to a BD PWM signal is also visible at the switching frequency of the BD PWM. In Fig. 7.6a, the tone at 882 Hz is less distinct and appears wider. This is because the measurement is done in LTspice, where the frequency resolution, i.e, spacing between measurement points, is dependent on the simulation time. Using a longer simulation time, the 882 Hz tone would look more distinct, but would take several hours to simulate. This does not affect the results, since some frequencies adjacent to the fundamental frequency are included as signal bins when calculating the SNR and SINAD. The other tones in Fig. 7.6 look more distinct, which is due to the scale of the frequency axis. At roughly 16 kHz, there is a slight increase in the spectrum, which then drops. This is caused by the RLC filter, which amplifies signals even before its resonance frequency.

Fig. 3.7 shows that the RLC filter has its resonance frequency around 23.2 kHz, and its effect can also be seen in Fig. 6.7. This also means that some frequencies in the range 16-20 kHz are amplified, including harmonics. Effort were made to mitigate this issue by increasing the cut-off frequency to 30 kHz and 46 kHz. The signal was affected negatively and introduced unwanted frequency components into the final signal, with slightly worse and unpredictable SNR and SINAD. Therefore, we opted to use the filter with a resonance peak at 23.2 kHz and a Q factor of 12.03.

Next, Fig. 7.7 depicts the THD+N in %, in comparison to the fundamental input signal frequency across the audible range (20 Hz–20 kHz). There is a peak in the graph close to 10 kHz. Frequencies up to 6.6 kHz have at least 3 distortion peaks in the audio range, increasing the distortion. Also, the second harmonic is increased according to the addition of dead time. The distortion in the frequency range of 16-20 kHz is amplified by the RLC filter as discussed earlier, contributing to the large spike of THD+N. Above 10 kHz, the distortion will appear at a frequency higher than 20 kHz. Therefore, they are not included in the calculation of THD+N, and the THD+N only consists of noise in this part of the graph. This causes the drop seen in Fig. 7.7. In other studies, such as [53], frequencies above 10 kHz are omitted for that reason. We decided to keep it to give an approximation of the noise floor and the THD+N across the whole audio spectrum.

Next is Fig. 7.8, which compares the SNR/SINAD to the input signal level, measured in dBFS. The amplifier is overloaded at around 0 dB, and has its maximum SNR and SINAD around -11 dBFS. The SNR and SINAD are both lowered when reducing the input signal level. The last test related to the frequency analysis is seen in Fig. 7.9, which compares the THD/THD+N with the output RMS voltage. The maximum voltage is the RMS output voltage of the maximum output voltage of our microcontroller, which is roughly 3.3 V. The lowest THD+N was found to be 0.15% at 0.36 V_{RMS}. This would be the input level that would generate the least amount of distortion.

8.2 Efficiency & Power consumption

This thesis achieved an apparent power efficiency of 86.4%. Comparably, the other works are somewhat more efficient, between 90 and 96%. However, the most important part is not the apparent power efficiency, but the power consumed. Especially the quiescent power losses that occur when the device is idle. For our application, it was measured to be 0.795 mW, which is significantly lower than that of the other devices in Table 8.1. For a battery-powered device, the power consumption should be as low as possible to achieve a high battery life. Therefore, it is very positive that the idle power consumption is low. The amplifier is still very efficient, meaning that most of the power supplied to it is used to create sound.

As can be seen in Fig. 7.11, the maximum efficiency is given at 2.66 V_{o} (V_{RMS}), which slowly decreases with output voltage. This means it is efficient in amplifying signals at a high input signal level, or high dBFS. However, comparing this to Fig. 7.10, the amplifier uses significantly more power above -10 dBFS. Even if it is more efficient, it will drain more power from the battery and thus lower battery life. Therefore, it would benefit the battery life to run the amplifier at a low input signal level, as long as it produces an appropriate sound quality and an acceptable level of distortion and noise.

9

Conclusion

I. A second-order $\Delta\Sigma$ modulator was designed in MATLAB and C, and tested in MATLAB. It could be implemented for an i^2s signal with correct input handling.
II. An efficient method was implemented to transform $\Delta\Sigma$ into BD PWM for simulation using LTspice. A BD PWM signal was generated in MATLAB, and the BD PWM was set up on the microcontroller. The resulting waveform was tested in a simulation environment but not on hardware.
III. The PCB was designed, and prototypes were printed. Due to soldering the PCB was deemed out of scope, the PCB wasn't soldered. Therefore, no actual testing could be done on the physical circuit.
IV. The second-order $\Delta\Sigma$ modulator was optimized to achieve an SNR of 75.5820 at 0 dBFS. In comparison to the maximum theoretical SNR, which is 76.1597 at 0 dBFS. The BD PWM signal generated in Matlab achieved an SNR of 79.3737 dB at 0 dBFS.
V. The lowest possible dead time was calculated to be 11 ns, and also gave the best results, avoiding shoot-through current and minimizing dead time.
VI. Simulation resulted in a THD+N Peak of 0.15 % at 1.323 kHz.
VII. The quiescent power consumption was measured to 0.8594 mW.
VIII. An apparent efficiency was measured to 86.43 %.
IX. Recommendations for future work are given in section 9.1.

To answer the research question, we do believe that it is feasible to implement a microcontroller-based digital Class D power amplifier using discrete components for piezoelectric actuators in bone conduction hearing aids.

9.1 Proposed System Overview for Future Work

This chapter outlines future work that could improve the results of this thesis. These include both changes to the work implemented and additional work to complete the system.

The microcontroller is the core of the system, used to process incoming audio data and control the output stage. The work was started, although it had to be abandoned, since it was deemed out of scope. A second order $\Delta\Sigma$ modulator was implemented, and PWM generation was successful for one output. The next step is to implement a program that can process incoming sound, preferably through Integrated Inter-IC Sound Bus (I^2S) signal lines. An efficient way of handling the audio stream is through so-called Ping-Pong-, or double- buffers. Next, multiple PWM signals must be created at the correct frequencies, and even more crucially, dead-time control between the outputs. This is feasible with our chosen microcontroller, which is capable of operating at a clock frequency of 150 MHz. Given that the MOSFETs have a maximum switch time of around 10 ns, a dead time of 15 ns is achievable. With a core frequency of $1/(15 \text{ ns}) = 66.67$ MHz, this microcontroller can effectively manage the dead time control. Subsequently, optimizing and minimizing the dead time and potentially adding an adaptive dead time control to avoid as much unwanted total harmonic distortion as possible is an important part of future work.

Secondly, the PCB needs to be thoroughly tested and if, necessary, improved. Adding discrete gate driver circuitry would be highly beneficial, as it would increase the gate current to the MOSFET, thereby speeding up the charging and discharging of the gate charge, reducing the switching times. Currently, we are combining dual outputs from our microcontroller GPIO, each providing a maximum current output of 4 mA, resulting in a total current of 8 mA. However, this current is not optimal. A current of around 1 mA would be more efficient, as it charges the gate charge much faster. One example is adding a low-side bipolar non-inverting totem-pole gate driver for the low side. For the high side, utilize some sort of gate driver circuitry that simply amplifies the current and ensures that the PMOS conducts when a low signal is present. This gate driver circuitry could also be implemented with MOSFETs, but then more parameters need to be taken into consideration such as inverting the PWM signal.

Next, as briefly discussed in Chapter 8, the $\Delta\Sigma$ modulator could be improved. For example, a higher-order and more efficient $\Delta\Sigma$ modulator incorporating error-feedback structure could prove useful, which would ensure better SNR at a lower PWM frequency. It would also be interesting to simulate the $\Delta\Sigma$ together with dithering and evaluate the potential improvements. The interpolation stage of the modulator could also be enhanced, either with a FIR filter, a cascaded integrator-comb (CIC) filter, or by implementing a phase-decomposition approach which decreases the multiplications needed substantially.

Another interesting continuation of the project would be to evaluate the effect of using Gallium Nitride (GaN) based switching power transistors instead of MOSFETs, in the H-bridge.

Bibliography

[1] Pixabay User (GDJ). *Musiknoter - Vinka Musik Abstrakt.* Free for use under Pixabay Content License. 2023. URL: `https://pixabay.com/sv/vectors/musiknoter-vinka-musik-abstrakt-8135228/` (visited on 02/13/2025).

[2] B.Derrickson and J.Tortora. *Introduction To The Human Body, 11th edition.* John Wiley Sons, 2019. Chap. 12.6. ISBN: 978111958546.

[3] J-E.Hawkins. *The physiology of hearing.* Encyclopædia Britannica. 2020. URL: `https://www.britannica.com/science/ear/The-physiology-of-hearing` (visited on 02/12/2024).

[4] T. Zahnert. “The differential diagnosis of hearing loss”. In: *Deutsches Ärzteblatt international* 108.25 (2011), p. 433. URL: `https://pmc.ncbi.nlm.nih.gov/articles/PMC3139416/` (visited on 03/15/2024).

[5] Better Health Channel. *Hearing Loss - How It Affects People.* n.d. URL: `https://www.betterhealth.vic.gov.au/health/conditionsandtreatments/hearing-loss-how-it-affects-people` (visited on 04/02/2025).

[6] *Hearing Loss in Adults.* American Speech-Language-Hearing Association, 2024. URL: `https://www.asha.org/public/hearing/Conductive-Hearing-Loss/` (visited on 01/15/2024).

[7] S. Ellsperman et al. “Review of bone conduction hearing devices”. In: *Audiology research* 11.2 (2021), pp. 207–219. URL: `https://pmc.ncbi.nlm.nih.gov/articles/PMC8161441/` (visited on 03/10/2024).

[8] E. Cumpston and P. Chen. *Implantable Hearing Devices.* Updated 2023. Treasure Island (FL): StatPearls Publishing, July 2023. URL: `https://www.ncbi.nlm.nih.gov/books/NBK578178/` (visited on 02/11/2024).

[9] T. GA Calon et al. “Minimally invasive ponto surgery versus the linear incision technique with soft tissue preservation for bone conduction hearing implants: a multicenter randomized controlled trial”. In: *Otology & neurotology* 39.7 (2018), pp. 882–893. URL: `https://pubmed.ncbi.nlm.nih.gov/29995008/` (visited on 02/11/2024).

[10] K. Robinette et al. “Transcutaneous versus percutaneous bone-anchored hearing aids: A quality of life comparison”. In: *American journal of otolaryngology* 44.2 (2023), p. 103758. URL: `https://pubmed.ncbi.nlm.nih.gov/36610247/` (visited on 01/11/2024).

[11] N.Shoman et al. “Comparison of passive versus active transcutaneous bone anchored hearing devices in the pediatric population”. In: *Journal of Otolaryngology-Head & Neck Surgery* 51.1 (2022), p. 44. URL: `https://pubmed.ncbi.nlm.nih.gov/36371309/` (visited on 02/02/2024).

[12] A.I. Colli-Menchi et al. *Design techniques for integrated CMOS class-D audio amplifiers*. World Scientific Publishing Company, 2016. ISBN: 9814704261, 9789814704267.

[13] J. Sang-Soo et al. “A Compact and Low-Cost MEMS Loudspeaker for Digital Hearing Aids”. In: *IEEE Transactions on Biomedical Circuits and Systems* 3.5 (2009), pp. 348–358. DOI: `10.1109/TBCAS.2009.2026429`. (Visited on 03/11/2024).

[14] B. Håkansson et al. “Electro-acoustic performance of the new bone vibrator Radioear B81: A comparison with the conventional Radioear B71”. In: *Int J Audiol* 54.5 (May 2015). Epub 2014 Dec 18, pp. 334–340. DOI: `10.3109/14992027.2014.980521`. (Visited on 02/15/2024).

[15] K.Hye Jin et al. “Improvement of low-frequency characteristics of piezoelectric speakers based on acoustic diaphragms”. In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 59.9 (2012), pp. 2027–2035. DOI: `10.1109/TUFFC.2012.2423`. (Visited on 02/15/2024).

[16] A.I. Colli-Menchi and E. Sánchez-Sinencio. “A High-Efficiency Self-Oscillating Class-D Amplifier for Piezoelectric Speakers”. In: *IEEE Transactions on Power Electronics* 30.9 (2015), pp. 5125–5135. DOI: `10.1109/TPEL.2014.2363406`. (Visited on 04/12/2024).

[17] Bob Cordell. *Designing Audio Power Amplifiers*. Routledge, 2010. Chap. 28. ISBN: 9781138555440.

[18] W. Kester. “Understand SINAD, ENOB, SNR, THD, THD+ N, and SFDR so you don’t get lost in the noise floor”. In: *MT-003 tutorial* (2009), pp. 01–08. URL: `https://www.researchgate.net/profile/Ramakrishna-Rssm/post/How-to-calculate-the-dynamic-range-of-an-ADC/attachment/59d6459579197b80779a0af3/AS%3A453975662174208%401485247640061/download/MT-003.pdf` (visited on 01/13/2025).

[19] Cochlear Limited. *Datasheet: Cochlear™ Baha® 6 Max Sound Processor, Cochlear Baha Connect and Attract System*. Accessed: 2025-05-20. 2022. URL: `https://mss-p-007-delivery.sitecorecontenthub.cloud/api/public/content/BUN871-Baha-6-Max-DataSheet.pdf` (visited on 08/11/2024).

[20] Oticon A/S. *Sentio 1 Mini Product Information Datasheet*. Accessed: 2025-05-20. 2024. URL: `https://p3.aprimocdn.net/dgs/32591f69-88ae-407a-8f3a-b192014bda3d/275144en_PI_Sentio%201%20Mini_Version%20A_2024.04_Original%20file.pdf` (visited on 08/13/2024).

[21] C. Palmer and L. Zitelli. “Chapter 2: My Hearing Aid Isn’t Working Like It Used to. . . ” In: *Seminars in Hearing* 45.2 (May 2024), pp. 153–171. DOI: `10.1055/s-0044-1786523`. (Visited on 04/12/2024).

[22] B.Andreycak. *Practical Considerations in High Performance MOSFET, IGBT and MCT Gate Drive Circuits*. Tech. rep. SLUA105. Texas Instruments, 1997. URL: `https://www.ti.com/lit/an/slua105/slua105.pdf` (visited on 03/15/2024).

[23] S. Mei et al. “The Class D Audio Power Amplifier: A Review”. In: *Electronics* 11.19 (2022). ISSN: 2079-9292. DOI: `10.3390/electronics11193244`. URL: `https://www.mdpi.com/2079-9292/11/19/3244` (visited on 05/15/2024).

[24] Bumman Kim. “Chapter 2 - Power amplifier fundamentals”. In: *RF and mm-Wave Power Generation in Silicon*. Ed. by K. Sengupta H. Wang. Oxford: Academic Press, 2016, pp. 17–58. ISBN: 978-0-12-408052-2. DOI: `https://doi.org/10.1016/B978-0-12-408052-2.00002-5`. (Visited on 02/15/2024).

[25] D. Oguzhan and O. Serhan. “An Overview of Power Amplifiers”. In: *Proceedings of the 5th International Mediterranean Science and Engineering Congress (IMSEC)*. Oct. 2020, pp. 124–129. URL: `https://d1wqtxts1xzle7.cloudfront.net/64794670/An_Overview_of_Power_Amplifiers-libre.pdf?1603921338=&response-content-disposition=inline%3B+filename%3DAn_Overview_of_Power_Amplifiers.pdf&Expires=1752219059&Signature=WS-OvqYBSSoqyApAHw6IhCCH7rJKlw9hIGGSOE-qdWrpCi6Y5kSBlx~mTZyWZRiVaj-JCQ6EdOjKfx2AvoSZNQQ3yLccyp0O72XwL9ur~h4XMlpvZykDkyQse2-tjI4QtEqYVoRB4tZ7-FXcsk3BVJeXneWg3vs6Ube1SGKiy2LB9qRUHuODweSEv9ltBnSjLg9t-Yj6ximmABt52DMPhBqFYChHTKc5HOiIRAYVcm5oXtgrqiN83It4g3ZViSFe2oLK-JjRl1pHkwF9N2TuVtYtCs5J2Vm8HrIqw__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA` (visited on 02/19/2024).

[26] M.Bloechl et al. “Class D Switching Power Amplifiers: Theory, Design, and Performance”. In: Apr. 2004, pp. 123–146. ISBN: 0-7803-8368-0. DOI: `10.1109/SECON.2004.1287908`. URL: `https://ieeexplore.ieee.org/document/1287908/` (visited on 02/11/2024).

[27] I. Bellili et al. “Low Power Class D Audio Amplifier with High Performance and High Efficiency”. In: *2019 6th International Conference on Image and Signal Processing and their Applications (ISPA)*. 2019, pp. 1–4. DOI: `10.1109/ISPA48434.2019.8966832`. (Visited on 02/15/2024).

[28] K. Nielsen. “A Review and Comparison of Pulse-Width Modulation (PWM) Methods for Analog and Digital Input Switching Power Amplifiers”. In: *Journal of The Audio Engineering Society* (1997). URL: `https://api.semanticscholar.org/CorpusID:106670535` (visited on 02/13/2024).

[29] L.Balogh. “Design And Application Guide For High Speed MOSFET Gate Drive Circuits By”. In: 2006. URL: `https://api.semanticscholar.org/CorpusID:14653623` (visited on 04/13/2024).

[30] V. Barkhordarian. “Power MOSFET Basics”. In: *International Rectifier* (2001). URL: `https://www.infineon.com/dgdl/Infineon-power_mosfet_basics-Article-v01_00-EN.pdf?fileId=8ac78c8c8d2fe47b018e625961741a0e` (visited on 04/25/2024).

[31] J. Adams and J. Honda. “Class D Audio Amplifier Basics”. In: *Application Note AN-1071* (2005). URL: `https://www.infineon.com/dgdl/an-1071.pdf?fileId=5546d462533600a40153559538eb0ff1` (visited on 01/12/2024).

[32] V4711. *RLC series circuit v2*. Licensed under CC BY-SA 3.0. Retrieved March 6, 2024. 2007. URL: https://commons.wikimedia.org/wiki/File:RLC_series_circuit_v2.svg.

[33] B. Karlström. *Kretsanalys*. Studentlitteratur, 2017,2018. Chap. 10. ISBN: 9789144125725.

[34] Lars Bengtsson. *Elektriska mätsystem och mätmetoder*. Wiley, 2015. Chap. 4. ISBN: 9789144080680.

[35] R. Schreier et al. *Understanding Delta-Sigma Data Converters*. Wiley, 2017. Chap. 13,14. ISBN: 9781119258278. URL: https://ieeexplore.ieee.org/servlet/wayf.jsp?entityId=https%3A%2F%2Fidp.chalmers.se%2Fopenathens&url=https%3A%2F%2Fieeexplore.ieee.org%2Fservlet%2Fopac%3Fbknumber%3D7906178 (visited on 12/24/2023).

[36] J. Reiss. "Understanding Sigma-Delta Modulation: The Solved and Unsolved Issues". In: *AES: Journal of the Audio Engineering Society* 56.1/2 (Jan. 2008), pp. 49–64. URL: https://aes2.org/publications/elibrary-page/?id=14375 (visited on 05/10/2024).

[37] B. Karlström and L. Bengtsson. *Transformer och filter*. Studentlitteratur, 2017. Chap. 8.3. ISBN: 9789144115368.

[38] F. Tabassum et al. "Comparison of FIR and IIR Filter Bank in Reconstruction of Speech Signal". In: *International Journal of Computer Science and Information Security* 14.9 (2016), p. 864. URL: https://www.researchgate.net/profile/Md-Islam-757/publication/340004096_Comparison_of_FIR_and_IIR_Filter_Bank_in_Reconstruction_of_Speech_Signal/links/5e723e5f299bf15718476d0a/Comparison-of-FIR-and-IIR-Filter-Bank-in-Reconstruction-of-Speech-Signal.pdf.

[39] R. Crochiere and L. Rabiner. "Optimum FIR digital filter implementations for decimation, interpolation, and narrow-band filtering". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 23.5 (1975), pp. 444–456. URL: https://ieeexplore.ieee.org/abstract/document/1162719/ (visited on 03/14/2024).

[40] J. A. E. P. van Engelen and R. J. van de Plassche. "New stability criteria for the design of low-pass sigma-delta modulators". In: *Proceedings of the 1997 International Symposium on Low Power Electronics and Design*. ISLPED '97. Monterey, California, USA: Association for Computing Machinery, 1997, pp. 114–118. ISBN: 0897919033. DOI: 10.1145/263272.263302. URL: https://doi.org/10.1145/263272.263302 (visited on 04/15/2024).

[41] F.J. Harris. "On the use of windows for harmonic analysis with the discrete Fourier transform". In: *Proceedings of the IEEE* 66.1 (1978), pp. 51–83. URL: https://ieeexplore.ieee.org/abstract/document/1455106/?casa_token=YZHFklbnW_oAAAAA:tKUY2iDB26Pv3HS_NBKH1jO7iPmE9fe2c6dbgkot7gfGmQXCnl6gdTW
9aOV4Y5GwySL1RCp9r (visited on 04/22/2024).

[42] Nexperia. *PMDXB290UNE 20 V, dual N-channel Trench MOSFET Product data sheet*. 19 January 2023. URL: https://assets.nexperia.com/documents/data-sheet/PMDXB290UNE.pdf (visited on 05/15/2024).

[43] Nexperia. *PMDXB950UPEL 20 V, dual P-channel Trench MOSFET Product data sheet.* 28 June 2016. URL: https://assets.nexperia.com/documents/data-sheet/PMDXB950UPEL.pdf (visited on 05/15/2024).

[44] Vishay Siliconix. *Complementary Low-Threshold MOSFET Pair SPICE Device Model Si1555DL Datasheet.* 28 June 2016. URL: https://www.vishay.com/docs/71495/71495.pdf (visited on 05/15/2024).

[45] NXP Semiconductors. *UM11126, LPC55S6x/LPC55S2x/LPC552x User manual.* 2023, Rev 2.6. URL: https://www.mouser.com/pdfDocs/NXP_LPC55S6x_UM.pdf?srsltid=AfmBOooR2hMlMXW6-sFyRhsn3_FAuH7EzUya9ztbxKh7I9FIEgWm__l9 (visited on 02/25/2024).

[46] E.Teofilo and J.Ogando. *UM11732,* $I^2sbusspecification, 3rded.$. 2019, Rev A2. URL: https://www.nxp.com/docs/en/user-manual/UM11732.pdf (visited on 02/25/2024).

[47] NXP Semiconductors. *UM11158 LPCXpresso55S69/55S28 Development Boards.* 2022, Rev 1.6. URL: https://www.mouser.com/pdfDocs/UM11158.pdf?srsltid=AfmBOopC3zqLK8TkYBhDN7f4GFhik-V58BiKmLCbQs4wmCS-XqiR4NSB (visited on 02/25/2024).

[48] S. Karmakar et al. “A 91 dB THD+N, Class-D Piezoelectric Speaker Driver Using Dual Voltage/Current Feedback for Resistor-Less LC Resonance Damping”. In: *IEEE Journal of Solid-State Circuits* 57.12 (2022), pp. 3726–3735. DOI: 10.1109/JSSC.2022.3207386. (Visited on 02/15/2024).

[49] S. Karmakar et al. “A Class-D Piezoelectric Speaker Driver Using A Quadrature Feedback Chopping Scheme achieving 29dB Large-Signal THD+N Improvement”. In: *2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits).* IEEE. 2023, pp. 1–2. DOI: \url{https://doi.org/10.23919/VLSITechnologyandCir57934.2023.10185166}. (Visited on 04/15/2024).

[50] Texas Instruments. *TPA2100P1: 19-Vpp Mono Class-D Audio Amplifier for Piezo/Ceramic Speakers.* Dec. 2008. URL: http://www.ti.com/lit/ds/symlink/tpa2100p1.pdf (visited on 02/13/2024).

[51] Texas Instruments. *DRV5825P Digital Input, 48VPP, 7.5 A Stereo, 15 A Mono, Piezo Speaker Driver With Adaptive I/V Limiter.* 2020. URL: https://www.ti.com/product/DRV5825P.

[52] New Japan Radio Co. *NJW1263 Ver.3.1_E: Analog Signal Input Class-D Amplifier for Piezo Speaker with DC-DC Converter.* Datasheet. URL: https://www.mouser.se/datasheet/2/294/NJW1263_E-779937.pdf (visited on 02/10/2024).

[53] Shoubhik Karmakar et al. “A- 91 dB THD+ N, Class-D Piezoelectric Speaker Driver Using Dual Voltage/Current Feedback for Resistor-Less LC Resonance Damping”. In: *IEEE Journal of Solid-State Circuits* 57.12 (2022), pp. 3726–3735. URL: https://ieeexplore.ieee.org/abstract/document/9906973 (visited on 07/13/2024).

A LTspice Netlist

```
* Generated by LTspice 24.1.10 for Windows.
V2 N002 0 3.3 Rser=1
V3 NMOS2 0 PWL file= Rser={RDRV}
V4 NMOS1 0 PWL file= Rser={RDRV}
M1 OUTA PMOS1 VrailA VrailA Si1555DL_P
M3 OUTA NMOS1 0 0 Si1555DL_N
M5 OUTB PMOS2 VrailB VrailB Si1555DL_P
M2 OUTB NMOS2 0 0 Si1555DL_N
C2 0 N005 100µ
C3 0 N006 10µ
V6 PMOS1 0 PWL file= Rser={RDRV}
V5 PMOS2 0 PWL file= Rser={RDRV}
R3 N002 N005 0.5
R4 VrailA N002 {RSUPPLY}
R5 N002 N006 1m
V1 N001 0 3.3 Rser=1
C4 0 N003 100µ
C5 0 N004 10µ
R6 N001 N003 0.5
R7 VrailB N001 {RSUPPLY}
R8 N001 N004 1m
R10 N007 OUTA 0.245
L1 OUTC N008 47µ
L2 N007 OUTD 47µ
C1 OUTC OUTD 1µ Rser=10m
R11 OUTB N008 0.245
B3 OUTRLC 0 V=V(OUTC) - V(OUTD)
.model NMOS NMOS
.model PMOS PMOS
.lib C:\Users\ka.emijo33\AppData\Local\LTspice\lib\cmp\standard.mos
.OPTIONS maxord=1
.OPTIONS itl1=1000
.OPTIONS itl2=1000
.OPTIONS itl4=1000
.OPTIONS numdgt=99
.OPTIONS measdgt=99
.OPTIONS gmin=1e-15
.OPTIONS abstol=1e-15
* tantalum
* ceramic
* routing
.OPTIONS plotwinsize=0
.param RDRV=10m RSUPPLY=10m
.four 4410 10 100 V(OUTRLC)
```

```
.tran 0 15.8766m 0 1n
.backanno
.end
```

B MATLAB SNRTest Function

```matlab

function [snr] = SNRTest(Output,Fs,f0)

% SNRTest - Computes the Signal-to-Noise Ratio (SNR) of a signal.
%
% INPUTS:
%   Output - Time-domain signal (vector)
%   Fs     - Sampling frequency (Hz)
%   f0     - Fundamental frequency of the signal (Hz)
%
% OUTPUT:
%   snr    - Signal-to-Noise Ratio in dB
%
% This function performs an FFT-based analysis of the input signal,
% identifies the signal bins, noise bins, and harmonic distortion components,
% and then computes the SNR. It also generates several diagnostic plots
% showing the power spectrum and amplitude spectrum.

% -------------------------------------------------------------------------
%  FFT and Windowing
% -------------------------------------------------------------------------
n_s0= length(Output);             % Length of the input signal
nfft = n_s0;                      % FFT length (same as signal length)
win = hann(n_s0)';                % Hann window for spectral analysis
                                  % Must transpose to match signal size
temp = fft(Output.*win,nfft);     % FFT of windowed signal

% Frequency vector for single-sided spectrum
f = (0:nfft/2) / nfft * Fs;

% Normalization factors for power calculation
amp_norm    = sum(win);           % Amplitude normalization (window sum)
power_norm = sum(win.^2) * n_s0;  % Power normalization

% Compute power spectrum (magnitude squared of FFT)
power_spec = abs(temp).^2;
power_spec = power_spec(1:n_s0/2+1);        % Single-sided spectrum
```

```
power_spec(2:end-1) = power_spec(2:end-1)*2;% Double non-DC and non-Nyquist
power_spec = power_spec / power_norm;       % Normalize by window power

% Time and frequency domain power (check values)
time_domain_power = sqrt(mean(Output.^2));
freq_domain_power = sqrt(sum(power_spec));

% Half-sided FFT versions for plotting
temp_halfsided  = temp(1:n_s0/2+1).*2;
temp_halfsided1 = temp(1:n_s0/2+1)/amp_norm;
temp_halfsided1(2:end-1) = temp_halfsided1(2:end-1)*2;

% -------------------------------------------------------------------------
%  Identify Signal and Noise Bins
% -------------------------------------------------------------------------
N=length(temp); % FFT length (used for bin indexing)

% Select frequency bins around the fundamental frequency f0
if f0<44.1*2
    signalBins = [BinIndex(N,(f0)-1*44.1,Fs) : BinIndex(N,(f0+1)+1*44.1*2,Fs)];
else
    signalBins = [BinIndex(N,(f0+1)-1*44.1*2,Fs) : BinIndex(N,(f0+1)+1*44.1*2,Fs)];
end

% Noise bins: up to 20 kHz (excluding signal bins later)
noiseBins = [1:BinIndex(length(temp), 20000, Fs)];

% Keep bins only within spectrum range
signalBins = signalBins(signalBins > 0 & signalBins <= length(power_spec));
noiseBins  = noiseBins(noiseBins > 0 & noiseBins <= length(power_spec));

% Remove overlap: ensure noise bins dont contain signal bins
noiseBins = setdiff(noiseBins, signalBins);

% -------------------------------------------------------------------------
%  Harmonic Distortion Analysis
% -------------------------------------------------------------------------
harmonic_indices = []; % Initialize list of harmonic bin indices

% Identify harmonic bins up to ~50th harmonic, limited to 20 kHz
for i = 2:50
    if f0*i >20000
```

```
        break;
    end
    harmonic_idx = BinIndex(N, f0 * i + 1, Fs);
    harmonic_idx = [harmonic_idx-1, harmonic_idx, harmonic_idx+1];
    harmonic_indices = [harmonic_indices, harmonic_idx];
end

% Exclude fundamental bins
harmonic_indices = setdiff(harmonic_indices, signalBins);

% Harmonic power (already squared, no need to square again)
harmonic_power = sum(power_spec(harmonic_indices));

% -------------------------------------------------------------------------
% Compute SNR
% -------------------------------------------------------------------------
s = sum(power_spec(signalBins));     % Signal power
noise = sum(power_spec(noiseBins));  % Noise power
n = abs(noise - harmonic_power);     % Exclude harmonic contribution

% SNR in dB (scaled by factor 1.5 per FFT conventions)
snr = 10 * log10(s / n);

% -------------------------------------------------------------------------
%  Convert to dB for Plotting
% -------------------------------------------------------------------------
power_spec_db   = 10*log10(abs(power_spec)/max(Output)); % Spectrum in dB
signalBins_db   = 10*log10(abs(power_spec(signalBins))/max(Output));
noiseBins_db    = 10*log10(abs(power_spec(noiseBins))/max(Output));

% -------------------------------------------------------------------------
%  Plot - Power Spectrum with Signal and Noise Bins
% -------------------------------------------------------------------------
figure;
plot(f, power_spec_db, 'LineWidth', 2.2); % Full spectrum
hold on;
stem(f(signalBins), signalBins_db, 'LineWidth', 2.2); % Signal bins
stem(f(noiseBins), noiseBins_db, 'LineWidth', 2.2);   % Noise bins

xlabel('Frequency (Hz)', 'FontSize', 38);
ylabel('Power (dB)', 'FontSize', 38);
```

```
title('Power Spectrum', 'FontSize', 30);
legend('Power Spectrum', 'Signal Bins', 'Noise Bins', 'FontSize', 22);
set(gca, 'FontSize', 26);
grid on;
xlim([f0-1000 f0+1000]);
hold off;

% Print computed values
disp(['Signal Power: ', num2str(s)]);
disp(['Noise Power: ', num2str(n)]);
disp(['Signal-to-Noise Ratio (SNR): ', num2str(snr), ' dB']);

% -------------------------------------------------------------------------
% Plot - Power Spectrum (Linear Scale)
% -------------------------------------------------------------------------
figure;
plot(f, abs(power_spec/max(Output)),'LineWidth',2.2);
hold on;
stem(f(signalBins), abs(power_spec(signalBins)/max(Output)));
stem(f(noiseBins), abs(power_spec(noiseBins)/max(Output)));

xlabel('Frequency (Hz)','FontSize',38);
ylabel('Power (dB)','FontSize',38);
title('Power Spectrum','FontSize',30);
legend('Power Spectrum', 'Signal Bins', 'Noise Bins','FontSize',22);
set(gca, 'FontSize', 26);
grid on
xlim([f0-1000 f0+1000]);
hold off;

% Reprint values
disp(['Signal Power: ', num2str(s)]);
disp(['Noise Power: ', num2str(n)]);
disp(['Signal-to-Noise Ratio (SNR): ', num2str(snr), ' dB']);

% -------------------------------------------------------------------------
%  Plot - Single-Sided Amplitude Spectrum
% -------------------------------------------------------------------------
figure;
specdb=20*log10(abs((temp_halfsided1)));
plot(linspace(0, 1, n_s0/2+1), specdb, 'LineWidth', 2.2);

title('Single-Sided Amplitude Spectrum in dB', 'FontSize', 18);
xlabel('Normalized Frequency (0 to 1)', 'FontSize', 18);
ylabel('Amplitude (dB)', 'FontSize', 18);
grid on;
```

```matlab

end
```

Listing 1: SNRTest function in MATLAB

C MATLAB SINADTest Function

```matlab

function [SINAD] = SINADTest(Output,Fs,f0)
% SINADTest - Computes the Signal-to-Noise and Distortion (SINAD) ratio of a signal.
%
% INPUTS:
%   Output - Time-domain signal (vector)
%   Fs     - Sampling frequency (Hz)
%   f0     - Fundamental frequency of the signal (Hz)
%
% OUTPUT:
%   SINAD    - Signal-to-Noise and Distortion Ratio in dB
%
% This function calculates the SINAD using FFT, identifies signal and
% noise bins, and generates plots of the power spectrum and single-sided amplitude
% spectrum for analysis.

% -------------------------------------------------------------------------
%  FFT and Windowing
% -------------------------------------------------------------------------
n_s0= length(Output);             % Length of input signal
nfft = n_s0;                      % FFT length
win = hann(n_s0)';                % Hann window; must transpose to match signal size
temp = fft(Output.*win,nfft);     % FFT of windowed signal

% Frequency vector for single-sided spectrum
f = (0:nfft/2) / nfft * Fs;

% Normalization factors
amp_norm   = sum(win);            % Amplitude normalization (sum of window)
power_norm = sum(win.^2) * n_s0;  % Power normalization (window energy)

% Compute power spectrum (single-sided)
power_spec = abs(temp).^2;
power_spec = power_spec(1:n_s0/2+1);          % Keep only positive frequencies
power_spec(2:end-1) = power_spec(2:end-1)*2;% Double non-DC, non-Nyquist
```

```
power_spec = power_spec / power_norm;       % Normalize by window power

% Compute RMS power in time and frequency domains (for reference)
time_domain_power = sqrt(mean(Output.^2));
freq_domain_power = sqrt(sum(power_spec));

% Half-sided FFT versions for plotting
temp_halfsided  = temp(1:n_s0/2+1).*2;
temp_halfsided1 = temp(1:n_s0/2+1)/amp_norm;
temp_halfsided1(2:end-1) = temp_halfsided1(2:end-1)*2;

% -------------------------------------------------------------------------
%  Identify Signal and Noise Bins
% -------------------------------------------------------------------------
N = length(temp); % FFT length (for bin indexing)

% Determine fundamental signal bins
if f0 < 44.1*2
    signalBins = [BinIndex(N,(f0)-1*44.1,Fs) : BinIndex(N,(f0+1)+1*44.1*2,Fs)];
else
    signalBins = [BinIndex(N,(f0+1)-1*44.1*2,Fs) : BinIndex(N,(f0+1)+1*44.1*2,Fs)];
end

% Noise bins (up to 20 kHz)
noiseBins = [1:BinIndex(length(temp), 20000, Fs)];

% Keep only bins within the valid spectrum
signalBins = signalBins(signalBins > 0 & signalBins <= length(power_spec));
noiseBins  = noiseBins(noiseBins > 0 & noiseBins <= length(power_spec));

% Remove any overlap: ensure noise bins do not include signal bins
noiseBins = setdiff(noiseBins, signalBins);

% -------------------------------------------------------------------------
%  Compute SINAD
% -------------------------------------------------------------------------
s = sum(power_spec(signalBins));     % Signal power
n = sum(power_spec(noiseBins));      % Noise power

% Compute SINAD in dB (divide by 1.5 per FFT scaling convention)
SINAD = 10 * log10(s / n);

%
```

```
-------------------------------------------------------------------------
% Convert to dB for plotting
% -------------------------------------------------------------------------
power_spec_db = 10*log10(abs(power_spec)/max(Output)); % Full spectrum in dB
signalBins_db = 10*log10(abs(power_spec(signalBins))/max(Output));
noiseBins_db  = 10*log10(abs(power_spec(noiseBins))/max(Output));

% -------------------------------------------------------------------------
%  Plot Power Spectrum (dB)
% -------------------------------------------------------------------------
figure;
plot(f, power_spec_db, 'LineWidth', 1.25); % Full spectrum
hold on;
stem(f(signalBins), signalBins_db, 'LineWidth', 1.25); % Signal bins
stem(f(noiseBins), noiseBins_db, 'LineWidth', 1.25);   % Noise bins

xlabel('Frequency (Hz)', 'FontSize', 18);
ylabel('Power (dB)', 'FontSize', 18);
title('Power Spectrum', 'FontSize', 18);
legend('Power Spectrum', 'Signal Bins', 'Noise Bins', 'FontSize', 18);
grid on;
xlim([f0-1000 f0+1000]);
hold off;

% Display computed values
disp(['Signal Power: ', num2str(s)]);
disp(['Noise Power: ', num2str(n)]);
disp(['Signal-to-Noise and Distortion Ratio (SINAD): ', num2str(snr), ' dB']);

% -------------------------------------------------------------------------
%  Plot Power Spectrum (Linear Scale)
% -------------------------------------------------------------------------
figure;
plot(f, abs(power_spec/max(Output)),'LineWidth',1.25); % Full spectrum
hold on;
stem(f(signalBins), abs(power_spec(signalBins)/max(Output))); % Signal bins
stem(f(noiseBins), abs(power_spec(noiseBins)/max(Output)));   % Noise bins

```

```
xlabel('Frequency (Hz)','FontSize',18);
ylabel('Power (dB)','FontSize',18);
title('Power Spectrum','FontSize',18);
legend('Power Spectrum', 'Signal Bins', 'Noise Bins','FontSize',18);
grid on
xlim([f0-1000 f0+1000]);
hold off;

% Repeat display of SINAD values
disp(['Signal Power: ', num2str(s)]);
disp(['Noise Power: ', num2str(n)]);
disp(['Signal-to-Noise and Distortion Ratio (SINAD): ', num2str(SINAD), ' dB']);

% ----------------------------------------------------------
% Plot Single-Sided Amplitude Spectrum
% ----------------------------------------------------------
figure;
specdb=20*log10(abs(temp_halfsided1));
plot(linspace(0, 1, n_s0/2+1), specdb, 'LineWidth', 1.25);

title('Single-Sided Amplitude Spectrum in dB', 'FontSize', 18);
xlabel('Normalized Frequency (0 to 1)', 'FontSize', 18);
ylabel('Amplitude (dB)', 'FontSize', 18);
grid on;

end
```

Listing 2: SINADTest function in MATLAB

D MATLAB Function for Halfband Design

```


function b = hbsynth(ntaps)
% hbsynth - Designs a halfband FIR filter using windowed sinc method.
%
% INPUT:
%   ntaps - Number of filter taps (must be odd)
%
% OUTPUT:
%   b     - Filter coefficients (FIR)
%
% Reference: Both understanding of Halfband filters and code taken from Neil Robertson, July 2025:
https://www.dsprelated.com/showarticle/1113.php


```

```
% -------------------------------------------------------------------------
%
% -------------------------------------------------------------------------
if mod(ntaps,2) == 0
    error('ntaps must be odd'); % Halfband filters require odd number of taps
end

% -------------------------------------------------------------------------
%  Define sample indices
% -------------------------------------------------------------------------
N = ntaps - 1;           % Filter order
n = -N/2:N/2;            % Symmetric sample indices around zero

% -------------------------------------------------------------------------
%  Truncated sinc function (ideal halfband impulse response)
% -------------------------------------------------------------------------
sinc = sin(n*pi/2) ./ (n*pi); % Standard sinc function for halfband
sinc(N/2 + 1) = 1/2;          % Correct the center sample (n=0) to 1/2

% -------------------------------------------------------------------------
%  Windowing
% -------------------------------------------------------------------------
win = hann(ntaps, "periodic");  % Hann window (periodic)

% -------------------------------------------------------------------------
% Apply window to truncated sinc
% -------------------------------------------------------------------------
b = sinc .* win'; % Multiply element-wise to shape the filter response

end
```

Listing 3: Halfband Design Function in MATLAB

E MATLAB Errorcreation Function

```

function [OutputVector, ErrorArray] = Errorcreation(RefSignal, bits)
% Errorcreation - Generates a symmetric quantization error array and output vector.
%
% INPUTS:
%   RefSignal - Reference signal (used to scale errors)
%   bits      - Number of quantization bits
%
% OUTPUTS:
%   OutputVector - Normalized output levels for quantization
%   ErrorArray   - Corresponding quantization error levels
%
% This function creates a symmetric error distribution for a given number of bits.

% ---------------------------------------------------------------------------
%  Compute basic parameters
% ---------------------------------------------------------------------------
levels = 2^(bits);            % Total number of quantization levels
Runlength = levels / 2;       % Half of the levels (for positive/negative symmetry)
maxi = max(RefSignal);        % Maximum value of the reference signal

% Initialize arrays
OutputVector = zeros(1, levels);
ErrorArray   = zeros(1, levels);

% ---------------------------------------------------------------------------
%  Fill error and output vectors
% ---------------------------------------------------------------------------
for i = 1:Runlength
    ErrorArray(levels) = 0;  % Ensure last element is zero

    % Assign symmetric error values
    ErrorArray(1, i*2-1) = i * maxi / Runlength;  % Positive error
    ErrorArray(1, i*2)   = -i * maxi / Runlength; % Negative error

    % Assign corresponding normalized output levels
    OutputVector(1, i*2-1) = i / levels;          % Positive output
    OutputVector(1, i*2)   = -i / levels;         % Negative output
end

```

```
%
    ---------------------------------------------------------------------
% Sort arrays
%
    ---------------------------------------------------------------------
OutputVector = sort(OutputVector);
ErrorArray   = sort(ErrorArray);

end
```

Listing 4: Errorcreation Function in MATLAB

F MATLAB Findindex Function

```


 function [ValueE, ValueO] = FindIndex(EArray, OArray, Input)
% FindIndex- Finds the closest value in a reference array and
    returns corresponding pair.
%
% INPUTS:
%   EArray - Reference array for comparison
%   OArray - Corresponding array of associated values
%   Input  - Input value to compare against EArray
%
% OUTPUTS:
%   ValueE - Closest value to Input in EArray
%   ValueO - Corresponding value from OArray
%
% This function computes the absolute difference between each
    element of EArray
% and the Input value, finds the index of the minimum difference,
    and returns
% the closest EArray value along with its associated OArray value.

%
    ---------------------------------------------------------------------
%  Compute absolute differences
%
    ---------------------------------------------------------------------
differences = abs(EArray - Input); % Element-wise difference
    between reference and input

%
    ---------------------------------------------------------------------
%  Find index of minimum difference
%
    ---------------------------------------------------------------------
```

```
[~, idx] = min(differences); % idx corresponds to the closest value
    in EArray

%
   ------------------------------------------------------------------------

%  Retrieve closest value and corresponding pair
%
   ------------------------------------------------------------------------

ValueE = EArray(idx); % Closest reference value
ValueO = OArray(idx); % Corresponding associated value

end
```

Listing 5: Findindex Function in MATLAB

G MATLAB Usampling Function

```


function Filtered_Signal = Upsampla(signal,OSR)
% UPSAMPLA - Multi-stage upsampling of a signal using halfband FIR
   filters.
%
% INPUTS:
%   signal - Input signal to be upsampled
%   OSR    - Target oversampling ratio (2, 4, 8, 16, 32, 64, 128,
   256)
%
% OUTPUT:
%   Filtered_Signal - Upsampled and filtered output signal
%
% This function implements multi-stage upsampling with factor 2 at
   each stage,
% using halfband FIR filters generated by hbsynth. It applies zero-
   phase
% filtering (filtfilt) at each stage and scales the output by 2 to
   compensate.

%
   ------------------------------------------------------------------------

%  Generate halfband filters for different stages
%
   ------------------------------------------------------------------------

b1 = hbsynth(51); % First-stage filter
b2 = hbsynth(15); % Second-stage filter
b3 = hbsynth(11); % Subsequent stages
OSR_step = 2;     % Each stage upsamples by factor of 2

%
   ------------------------------------------------------------------------
```

```
%  Apply multi-stage upsampling depending on OSR
%
    ------------------------------------------------------------------------
switch OSR
    case 256
        % Stage 1
        u1 = upsample(signal, OSR_step);            % Insert zeros
    between samples
        f1 = filtfilt(b1, 1, u1).*2;              % Filter and scale

        % Stage 2
        u2 = upsample(f1, OSR_step);
        f2 = filtfilt(b2, 1, u2).*2;

        % Stage 3
        u3 = upsample(f2, OSR_step);
        f3 = filtfilt(b3, 1, u3).*2;

        % Stage 4
        u4 = upsample(f3, OSR_step);
        f4 = filtfilt(b3, 1, u4).*2;

        % Stage 5
        u5 = upsample(f4, OSR_step);
        f5 = filtfilt(b3, 1, u5).*2;

        % Stage 6
        u6 = upsample(f5, OSR_step);
        f6 = filtfilt(b3, 1, u6).*2;

        % Stage 7
        u7 = upsample(f6, OSR_step);
        f7 = filtfilt(b3, 1, u7).*2;

        % Stage 8
        u8 = upsample(f7, OSR_step);
        Filtered_Signal = filtfilt(b3, 1, u8).*2;

     case 128
        % Similar stages for OSR = 128
        u1 = upsample(signal, OSR_step);
        f1 = filtfilt(b1, 1, u1).*2;

        u2 = upsample(f1, OSR_step);
        f2 = filtfilt(b2, 1, u2).*2;

        u3 = upsample(f2, OSR_step);
        f3 = filtfilt(b3, 1, u3).*2;

        u4 = upsample(f3, OSR_step);
        f4 = filtfilt(b3, 1, u4).*2;

        u5 = upsample(f4, OSR_step);
        f5 = filtfilt(b3, 1, u5).*2;
```

```

        u6 = upsample(f5, OSR_step);
        f6 = filtfilt(b3, 1, u6).*2;

        u7 = upsample(f6, OSR_step);
        Filtered_Signal = filtfilt(b3, 1, u7).*2;

    case 64
        % Stages for OSR = 64
        u1 = upsample(signal, OSR_step);
        f1 = filtfilt(b1, 1, u1).*2;

        u2 = upsample(f1, OSR_step);
        f2 = filtfilt(b2, 1, u2).*2;

        u3 = upsample(f2, OSR_step);
        f3 = filtfilt(b3, 1, u3).*2;

        u4 = upsample(f3, OSR_step);
        f4 = filtfilt(b3, 1, u4).*2;

        u5 = upsample(f4, OSR_step);
        f5 = filtfilt(b3, 1, u5).*2;

        u6 = upsample(f5, OSR_step);
        Filtered_Signal = filtfilt(b3, 1, u6).*2;

    case 32
        % Stages for OSR = 32
        u1 = upsample(signal, OSR_step);
        f1 = filtfilt(b1, 1, u1).*2;

        u2 = upsample(f1, OSR_step);
        f2 = filtfilt(b2, 1, u2).*2;

        u3 = upsample(f2, OSR_step);
        f3 = filtfilt(b3, 1, u3).*2;

        u4 = upsample(f3, OSR_step);
        f4 = filtfilt(b3, 1, u4).*2;

        u5 = upsample(f4, OSR_step);
        Filtered_Signal = filtfilt(b3, 1, u5).*2;

    case 16
        % Stages for OSR = 16
        u1 = upsample(signal, OSR_step);
        f1 = filtfilt(b1, 1, u1).*2;

        u2 = upsample(f1, OSR_step);
        f2 = filtfilt(b2, 1, u2).*2;

        u3 = upsample(f2, OSR_step);
        f3 = filtfilt(b3, 1, u3).*2;

        u4 = upsample(f3, OSR_step);
```

```
        Filtered_Signal = filtfilt(b3, 1, u4).*2;

    case 8
        % Stages for OSR = 8
        u1 = upsample(signal, OSR_step);
        f1 = filtfilt(b1, 1, u1).*2;

        u2 = upsample(f1, OSR_step);
        f2 = filtfilt(b2, 1, u2).*2;

        u3 = upsample(f2, OSR_step);
        Filtered_Signal = filtfilt(b3, 1, u3).*2;

    case 4
        % Stages for OSR = 4
        u1 = upsample(signal, OSR_step);
        f1 = filtfilt(b1, 1, u1).*2;

        u2 = upsample(f1, OSR_step);
        Filtered_Signal = filtfilt(b2, 1, u2).*2;

    case 2
        % Single stage for OSR = 2
        u1 = upsample(signal, OSR_step);
        Filtered_Signal = filtfilt(b1, 1, u1).*2;
end
```

Listing 6: Upsampling function in MATLAB

H MATLAB ΔΣ Modulator Function

```

function [O] = AccDeltaSigma(Signal, RefSignal, bits, order)
% ACCDeltaSigma - Implements a first- or second-order Delta-Sigma modulator.
%
% INPUTS:
%   Signal      - Input analog-like signal to be converted
%   RefSignal   - Reference signal used for quantization
%   bits        - Number of quantization bits
%   order       - Order of the Delta-Sigma modulator (1 or 2)
%
% OUTPUT:
%   O           - Output bitstream from the Delta-Sigma modulator
%
% This function implements an accumulator-based Delta-Sigma modulator using
% quantization levels defined by Errorcreation. It supports first- and second-order
% modulators with one or two accumulators, respectively.

% --------------------------------------------------------------------------
```

```
%  Generate quantization levels and errors
%  ------------------------------------------------------------------
[OA, EA] = Errorcreation(RefSignal, bits);
% OA: Output levels for quantization
% EA: Corresponding quantization errors

%  ------------------------------------------------------------------
%  Initialize vectors and accumulators
%  ------------------------------------------------------------------
O = zeros(size(Signal)); % Output vector
e = zeros(size(Signal)); % Error vector
accumulator1 = 0;        % First accumulator
accumulator2 = 0;        % Second accumulator (used for second-order)

%  ------------------------------------------------------------------
% Choose processing based on modulator order
%  ------------------------------------------------------------------
switch order
    case 1
        % First-order Delta-Sigma
        for i = 1:size(Signal, 2) % Process each sample in the input signal
            if i < 2
                % For first sample, initialize accumulator and output
                accumulator1 = Signal(1, i);
                e(1, i) = 0;
                O(1, i) = 0;
            else
                % Accumulate input minus previous quantization error
                accumulator1 = accumulator1 + (Signal(1, i) - e(1, i-1));

                % Quantize the accumulator and get corresponding output
                [e(1, i), O(1, i)] = FindIndex(EA, OA, accumulator1);
            end
        end

    case 2
        % Second-order Delta-Sigma
        for i = 1:size(Signal, 2)
            if i < 2
```

```
                % Initialize accumulators and output for first sample
                accumulator1 = Signal(1, i);
                e(1, i) = 0;
                O(1, i) = 0;
            else
                % First accumulator integrates input minus previous error
                accumulator1 = accumulator1 + (Signal(1, i) - e(1, i-1));

                % Second accumulator integrates first accumulator minus previous error
                accumulator2 = accumulator2 + (accumulator1 - e(1, i-1));

                % Quantize the second accumulator and get output
                [e(1, i), O(1, i)] = FindIndex(EA, OA, accumulator2);
            end
        end
end

end
```

Listing 7: ΔΣ Modulator Function in MATLAB

I MATLAB Function for Resampling Gate signals

```

 function reformat_for_LTspice(input_filename, output_filename)
% Reformat_for_LTspice - Reformats a voltage signal text file for LTSpice.
%
% This function reads a text file with time vs. voltage/magnitude values,
% identifies transitions (rising/falling edges), and keeps only the start
% and end points of each level to reduce the data size. The reformatted
% signal is saved in a new text file ready for use as a voltage source in LTSpice.
%
% INPUTS:
%   input_filename  - String, name of the input text file
%   output_filename - String, name of the output text file to save
%

%% Read input data
data = readmatrix(input_filename);  % Read the input file
time_vector = data(:, 1)';          % Time or x-axis values
magnitude_vector = data(:, 2)';     % Voltage/magnitude values
```

```matlab

%%  Identify transition points
% Rising edges: magnitude increases
Low_Last_index = find(diff(magnitude_vector) > 0);
% Falling edges: magnitude decreases
Low_first_index = find(diff(magnitude_vector) < 0) + 1;

% Combine all low-level indices and sort
Low_indexes = sort([Low_first_index, Low_Last_index]);

% Rising edges for high levels
High_last_index = find(diff(magnitude_vector) < 0);
High_first_index = find(diff(magnitude_vector) > 0) + 1;
High_first_index_1 = 1;  % Always include the first sample

% Combine all high-level indices and sort
High_indexes = sort([High_first_index_1, High_first_index,
    High_last_index]);

%%  Combine all indices to keep
new_signal_index = sort([Low_indexes, High_indexes]);

%%  Extract the reduced signal
new_time = time_vector(new_signal_index);            % Keep only
    selected time points
new_magnitude = magnitude_vector(new_signal_index);% Keep only
    selected voltage points

%%  Save to LTSpice-ready text file
LTSpice_signal = [new_time(1,:)' new_magnitude(1,:)']; % Combine as
     two-column matrix
writematrix(LTSpice_signal, output_filename);           % Write to
    text file

end
```

Listing 8: Resampling Gate Signals Function in MATLAB

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se